

Opinnäytetyö (AMK)

Elektroniikan koulutusohjelma

Elektroniikkasuunnittelu

2017

Sami Rautanen

SÄRÖMITTAPAIKAN AUTOMATISOINNIN SUUNNITTELU JA TOTEUTUS

Sami Rautanen

SÄRÖMITTAPAIKAN AUTOMATISOINNIN SUUNNITTELU JA TOTEUTUS

Työn tarkoituksena oli suunnitella ja toteuttaa siirtolinjojen automaattinen ohjausjärjestelmä kaapelitelevisioverkon laitteiden tuotannon särömittapaikalle. Tavoitteena oli automatisoida mittapaikka ja poistaa työn tekijältä turhaa aikaa kuluttavia toimia. Särömittapaikalla mitattavien laitteiden säröominaisuuksia testataan taajuuksilla 25—1 006 MHz, joten suunniteltavan ohjausjärjestelmän laajakaistaisuus aiheutti omat haasteensa sen suunnittelussa.

Järjestelmä toteutettiin suunnittelemalla piirilevy, jonka RF-kytkimiä ohjataan Arduinolla, joka saa käskynsä tietokoneelta. Laitteen ajuri toteutettiin LabVIEW-ohjelmistoilla, ja siitä pyrittiin tekemään yleiskäyttöinen, jotta tuotannossa olevat muut samankaltaiset ohjausjärjestelmät voisivat käyttää sitä.

Työn lopputuloksena saatiin ohjausjärjestelmä, joka ei saavuttanut kaikilta osin haluttuja laatuvaatimuksia, sillä porttien heijastusvaimennukset eivät olleet tarpeeksi hyviä. Siirtolinjojen automaattinen reititys kuitenkin onnistui hyvin ja järjestelmän vaimennus oli tarpeeksi pieni. Lisäksi järjestelmän ajuri toimi suunnitellusti.

Järjestelmän kehitystä jatketaan tämän työn jälkeen, jotta sille asetetut vaatimukset saavutetaan ja järjestelmä voidaan ottaa käyttöön tuotannossa.

ASIASANAT:

Arduino, LabVIEW, Lean, LINX, RF-kytkin, Siirtolinja, Särö

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme in Electronics

2017 | Total number of pages: 39

Instructor: Juha Nikkanen, Lic. Sc. (Tech.)

Sami Rautanen

DESIGNING AND IMPLEMENTING THE AUTOMATION FOR DISTORTION MEASUREMENTS

The purpose of this Bachelor's thesis was to design and implement an automation for distortion measurements of cable television networks devices. The goal was to have all of the distortion measurements automatized and dispose of the unprofitable time workers have to use to complete the measurements. Distortion measurements use frequency range 25—1 006 MHz, hence the broadband caused some challenges in designing the automation system.

The automation system was implemented designing a circuit board, whose RF-switches are controlled by Arduino which is in turn controlled by a computer. The systems driver was implemented using the LabVIEW system-design platform. The intention was to create universal driver for similar automation systems.

The result was an automation system which did not achieve all of the desired specifications. This was due to the poor return losses of the inputs and outputs. The automatic routing of the transmission lines was successful and the attenuation of the system was adequately small. In addition, the systems driver operated as planned.

The automation systems improvements will be continued after this thesis to achieve all of the desired specifications and to introduce the automation system in the production.

KEYWORDS:

Arduino, Distortion, LabVIEW, Lean, LINX, RF-switch, Transmission line

SISÄLTÖ

KÄYTETYT LYHENTEET

1 JOHDANTO	1
2 TYÖN TAVOITE JA TYÖSSÄ KÄYTETYT TYÖKALUT	2
2.1 Lean	3
2.2 LabVIEW ja LabVIEW Makerhub LINX -kirjasto	4
2.3 TestStand	5
2.4 Arduino	5
3 RF-PIIRISUUNNITTELU JA SÄRÖT	7
3.1 Siirtolinjat	7
3.2 Vastus, kondensaattori ja kela	8
3.3 Sovituspiirit	11
3.3.1 L-sovituspiiri	11
3.3.2 Π - ja T-sovituspiirit	13
3.4 Säröt	15
4 TYÖN SUUNNITTELU	17
4.1 RF-lohkokaavion suunnittelu	17
4.2 RF-kytkinten ohjausten suunnittelu ja Arduinon valinta	18
4.3 RF-kytkinten ohjausten ja käyttöjännitteiden suunnittelu	19
4.4 Komponenttien valinnat	20
4.4.1 RF-kytkin	20
4.4.2 Summain	21
4.4.3 Sovituspiiri	22
5 TYÖN TOTEUTUS	24
5.1 Piirin suunnittelu ja piirtäminen	24
5.2 Arduinon ohjaus tietokoneella	26
5.2.1 RF-kytkinten ohjaus	27
5.2.2 EEPROMin kirjoitus ja lukeminen	28
5.3 Ensimmäiset mittaukset	28
5.4 Moniveräjän parannuksia	31

5.5 Moniveräjän särömittaukset	33
5.6 Moniveräjän kotelointi ja koteloinnin vaikutus mittaustuloksiin	34
6 PÄÄTELMÄT	36
7 YHTEENVETO	37
LÄHTEET	38

LIITTEET

Liite 1. Moniveräjän loppumittausten tulokset

KUVAT

Kuva 1. Suunnitellun systeemin lohkokaavio.	2
Kuva 2. LabVIEW'lla toteutettu blink-ohjelma.	4
Kuva 3. Esimerkki Arduinosta, kuvassa Arduino Mega 2560.	6
Kuva 4. Signaaligeneraattori kytkettynä siirtolinjalla RC-piiriin. [11]	7
Kuva 5. Vastuksen sijaiskytkentä radiotaajuuksilla. [1]	8
Kuva 6. Woundwire-vastuksen impedanssin käyttäytyminen taajuuden kasvaessa. [1]	8
Kuva 7. Kondensaattorin sijaiskytkentä. [1]	9
Kuva 8. Kondensaattorin impedanssin käyttäytyminen taajuuden kasvaessa. [1]	9
Kuva 9. Kelan sijaiskytkentä. [1]	10
Kuva 10. Kelan impedanssin käyttäytyminen taajuuden kasvaessa. [1]	10
Kuva 11. Esimerkki alipäästöisestä L-sovituspieristä.	12
Kuva 12. Suuremman Q-arvon saamiseksi kaksi L-sovituspieriä kytkettynä sarjaan.	12
Kuva 13. Yhteenveto L-sovituspieristä. [1]	13
Kuva 14. Π -sovituspieri esitettynä kahden L-sovituspierin yhdistelmällä. [1]	14
Kuva 15. Π -sovituspieri, jossa kolme komponenttia. [1]	14
Kuva 16. T-sovituspieri, jossa kolme komponenttia. [1]	15
Kuva 17. Esimerkki vahvistimen säröominaisuuksista.	16
Kuva 18. Moniveräjän RF-lohkokaavio.	17
Kuva 19. Error-viestit yritettäessä käyttää Arduinon analogista pinniä LINUX-kirjaston avulla.	18
Kuva 20. Arduino Mega 2560:n testauksessa käytetty kytkentä.	19
Kuva 21. PE42722:n tavanomainen kytkentä. [18]	21
Kuva 22. Laskettavan sovituspierin komponentit.	23
Kuva 23. Ladottu moniveräjän piirilevy.	25
Kuva 24. Moniveräjän prototyyppi.	25
Kuva 25. Käytössä olevien digitaalisten lähtöjen taulun hakeminen LINUX Resource -klusterista.	26

Kuva 26. Ohjelman osa, jossa tarkastetaan Arduinon malli ja toteutetaan haluttu kytkentä.	27
Kuva 27. EEPROMin lukemiseen käytetty ali-VI.	28
Kuva 28. Käytetty mittausasetelma.	29
Kuva 29. Moniveräjän kotelon etuosa.	34

KUVIOT

Kuvio 1. Summaimen 3DSS2 heijastusvaimennus.	22
Kuvio 2. Summaimen 3DSS2 läpäisy.	22
Kuvio 3. Moniveräjän portin IN1 heijastusvaimennus S11 taajuuden funktiona.	29
Kuvio 4. Moniveräjän portin IN1 läpäisy taajuuden funktiona.	30
Kuvio 5. Moniveräjän portin IN1 heijastusvaimennus S22 taajuuden funktiona.	30
Kuvio 6. Moniveräjän portin IN1 parannusten vaikutus heijastusvaimennukseen.	31
Kuvio 7. Parannellun piirin portin IN1 heijastusvaimennus S11.	32
Kuvio 8. Parannellun piirin portin IN1 läpäisy.	32
Kuvio 9. Parannellun piirin portin IN1 heijastusvaimennus S22.	32
Kuvio 10. Mitatut CTB-säröt.	33
Kuvio 11. Mitatut CSO-säröt.	33
Kuvio 12. Koteloidun moniveräjän portin IN1 heijastusvaimennus S11.	34
Kuvio 13. Koteloidun moniveräjän portin IN1 läpäisy.	35
Kuvio 14. Koteloidun moniveräjän portin IN1 heijastusvaimennus S22.	35

KÄYTETYT LYHENTEET

Q	Quality factor, piirin hyvyysluku
CSO	Composite Second Order beat, yksi toisen kertaluokan säröistä
CTB	Composite Third order Beat, yksi kolmannen kertaluokan säröistä
DUT	Device Under Test, mitattavana oleva tuote
EEPROM	Electronically Erasable Programmable Read-Only Memory, haihtumaton muisti, jota käytetään pienehkölle datalle
GPIB	General Purpose Internal Bus, Hewlett Packardin kehittämä väylä, joka mahdollistaa kommunikoinnin usean laitteen välillä
HV	Heijastusvaimennus, siirtolinjan epäjatkuvuuskohdan aiheuttama vaimennus
IC	Integrated circuit, mikropiiri, johon integroitu aktiivisia ja/tai passiivisia komponentteja
LabVIEW	Laboratory Visual Instrument Engineering Workbench, National Instrumentsin kehittämä graafinen ohjelmointikieli
PAL	Phase Alternate Line, analogisen videokuvan värijärjestelmä ja koodausmenetelmä
RF	Radio frequency, radiotaajuus (3 kHz – 300 GHz)
USB	Universal serial bus, standardi sarjaliikennearkkitehtuuri tietokoneen ja oheislaitteen välillä
UART	Universal asynchronous receiver/transmitter, ohjelmoitu piiri, joka ohjaa sarjaliikenneväylää
VI	Virtual instrument, National Instrumentsin käyttämä nimitys LabVIEW:llä toteutetulle ohjelmalle

1 JOHDANTO

Teleste Oyj valmistaa kaapelitelevisioverkon laitteita, jotka käyttävät korkeita, yli 1 GHz taajuuksia. Valmistetuista tuotteista täytyy testata monia eri ominaisuuksia, jotta saadaan varmistettua niiden luotettava toimivuus. Säröominaisuudet ovat yksi tärkeä osa testattavista ominaisuuksista, mutta niiden mittaukset eivät vaatineet operaattorilta, eli mittaus-ten suorittajalta, muita toimenpiteitä, kuin vaihtaa DUT:n portteihin kytkettyjen kaapelien paikkoja. Mittaukset itsessään oli jo aiemmin automatisoitu, mutta särömittapaikat haluttiin automatisoida myös muilta osin. Niinpä tämän työn tavoitteeksi tuli luoda laite, joka osaa hoitaa siirtolinjojen reitityksen automaattisesti ja poistaa operaattorilta tarpeen siirrellä siirtolinjoja. Sen avulla operaattori pystyy tekemään tehokkaasti töitä ja keskittymään enemmän olennaisiin asioihin. Laitetta alettiin nimittää moniveräjäksi ja tässä työssä käytetään samaa nimitystä. Aiemmin käytössä oli Agilentin 87075C Multiportti, joka ei toiminut enää vaatimusten mukaisesti mitattavien tuotteiden toimintataajuuden ylärajan noustessa 1 GHz:stä 1,2 GHz:iin.

Mittattavat laitteet toimivat taajuuksilla 5—1 218 MHz, ja mittauksia suoritetaan särömittapaikalla taajuuksilla 25—1 006 joka loi omat haasteensa suunnitteluun. Signaali käyttäytyy komponenteissa eri tavalla eri taajuuksilla mm. hajasuureiden takia, esimerkiksi kondensaattori näkyy matalilla taajuuksilla kondensaattorina, mutta korkeilla taajuuksilla kelana. Siirtolinjan vaikutus signaaliin on myös erilainen eri taajuuksilla [1].

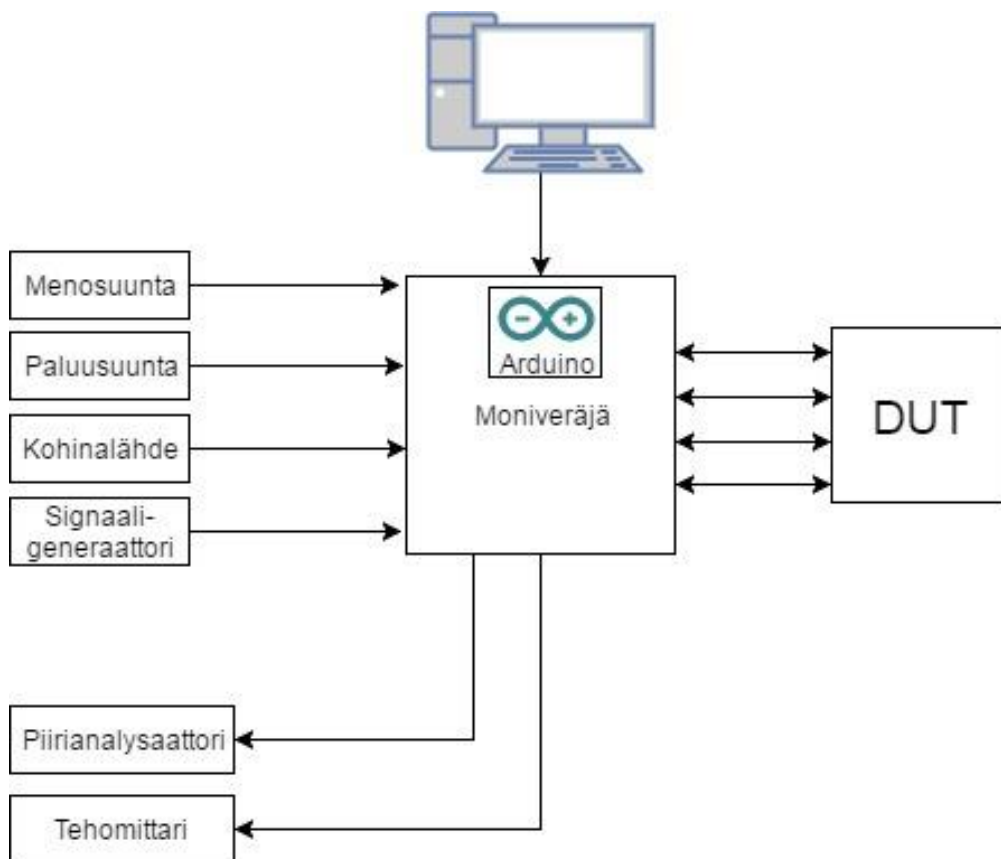
Työ toteutettiin käyttämällä jo ennestään käytössä olleita työkaluja. Moniveräjän piirilevy on suunniteltu Mentor Graphicsin Board Architect -piirisuunnitteluohjelmalla. Moniveräjän RF-kytkimiä ohjataan Arduino Mega 2560:llä, jolle käskyt annetaan tietokoneelta. Käskyt luodaan National Instrumentsin TestStand ja LabVIEW -kehitysympäristöillä toteutetuilla ohjelmilla.

Samankaltaisen työn on aiemmin tehnyt 2012 Petri Pulkkinen Kajaanin ammattikorkeakoulussa. Hänen työnsä aiheena oli EMC-laitteiston signaaliteiden ohjausyksikkö. Se toteutettiin RF-releillä, joita ohjataan LabVIEW'illä tehdyllä ohjelmalla. [2]

2 TYÖN TAVOITE JA TYÖSSÄ KÄYTETYT TYÖKALUT

Työn tavoitteena oli automatisoida särömittapaikka, jotta operaattori voi mittauksien aikana suorittaa muita tehtäviä. Särömittapaikan mittaukset oli automatisoitu jo aiemmin, mutta operaattori joutui kuitenkin siirtelemään siirtolinjoja mittausten aikana. Suunnitellulla moniveräjällä, joka hoitaa siirtolinjojen reitityksen automaattisesti, oli tarkoitus vapauttaa operaattorilta mittauksiin kuuluva aika.

Moniveräjää täytyy luonnollisesti ohjata samalla ohjelmalla, kuin mittauksia suoritetaan, jotta siirtolinjojen reititys tapahtuu oikeaan aikaan. Kuvassa 1 on suunnitellun systeemin lohkokaavio, josta selviää systeemin kokonaiskuva.



Kuva 1. Suunnitellun systeemin lohkokaavio.

Tietokoneella ohjataan Arduinoa, joka ohjaa moniveräjän RF-kytkimiä. Moniveräjälle tulee neljä lähdettä, signaaligeneraattori, kohinalähde, sekä meno- ja paluusuunnat, jotka ohjataan johonkin DUT:n tuloista. Halutun DUT:n lähdön moniveräjä ohjaa piirianalysaattorille

tai tehomittarille. Tavoitteena oli, että moniveräjä vaikuttaa mittaustuloksiin mahdollisimman vähän.

Työn aikana käytettiin monia erilaisia työkaluja ja ohjelmistoja: piirisuunnittelussa käytetty Mentor Graphicsin Board Architect -piirisuunnitteluohjelma, RF-kytkimien ohjauksissa käytetty Arduino mikro-ohjain sekä Arduinon ja tietokoneen välisessä kommunikoinnissa käytetyt LabVIEW ja TestStand -ohjelmistot. Työn tarkoitus pohjautuu lean-johtamisfilosofiaan, sillä pyritään lyhentämään tuotteiden lähimenoaikaa ja poistamaan hukkaa.

2.1 Lean

Lean-johtamisfilosofia on muodostettu autonvalmistaja Toyota Motor Corporationin tuotannossa käytetystä TPS (Toyota Production System) ajatusmallista. Siinä pyritään poistamaan työstä turhia osia (hukka), joka ei lisää työn arvoa tai laatua. Tällaisia arvoa tuottamattomia toimintoja tai työn osia ovat: ylituotanto, odottelu, tarpeeton kuljettelu, tarpeettomat varastot, tarpeeton liikkuminen, viat, ylikäsittely ja työntekijän luovuuden käyttämättä jättäminen. Tämän työn tarkoitus pohjautuu tarpeettoman liikkumisen, odottelun ja virheellisen käsittelyn poistamiseen. [3]

Kun signaaleita joutuu mittausten aikana muuttamaan, niin samalla on keskeytettävä työvaihe, jota on suorittamassa. Moniveräjä poistaa ongelman, jolloin operaattorin tarvitsee vain mittauksen aluksi asettaa mitattava laite paikoilleen ja yhdistää moniveräjään ja lopuksi, kun mittaukset ovat valmiit, irrottaa se.

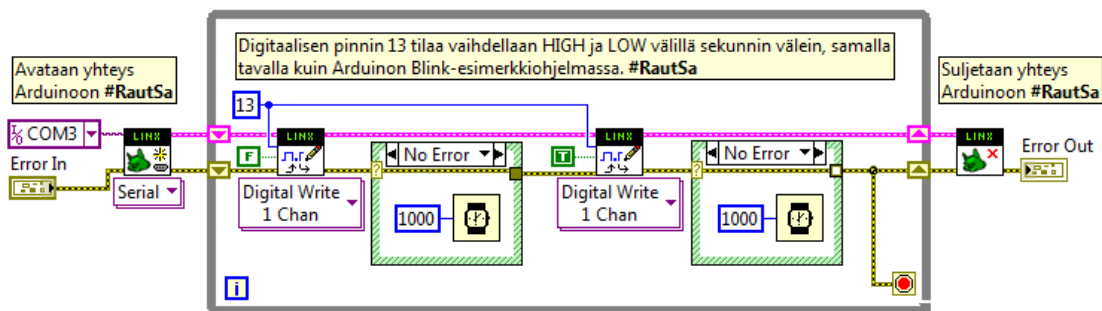
Virheellinen käsittely tarkoittaa, että työn suorittamiseksi työn tekijän on käytettävä aikaa tarpeettomien osien suoritukseen työn käsittelyssä, josta syntyy tarpeetonta liikkumista ja virheitä tuotteeseen. Tämä aiheuttaa hukkaa ja voi johtua esimerkiksi keinoista työkaluista tai puutteellisesta työn suunnittelusta [3]. Moniveräjä poistaa virheellistä käsittelyä, sillä se suorittaa automaattisesti oikean signaaliteiden reitityksen, jolloin suurin osa virhemahdollisuuksista häviää.

2.2 LabVIEW ja LabVIEW Makerhub LINX -kirjasto

LabVIEW on National Instrumentsin suunnittelema graafinen ohjelmointikieli, joka on suunniteltu helppokäyttöiseksi ja nopeasti opittavaksi. Tarkoituksena on kustannustehokkuus, valmiiden kirjastojen avulla hankalakin projekti valmistuu nopeasti ja pienellä vaivalla. LabVIEW:ä käyttävät insinöörit ja tutkijat ympäri maailman ja se on suunniteltu pitämään sisällään työkalut moneen eri käyttötarkoitukseen, muun muassa automaattisten testien tekemiseen, johon sitä tässä työssä on käytetty. [4], [5]

LabVIEW oli luonnollinen valinta ohjelmointikieleksi, sillä Telesten tuotannossa oli jo ennuudesta käytetty LabVIEW:tä ohjelmointikielenä. LabVIEW:llä tehdyistä ohjelmista on helppo tehdä modulaarisia, joten koodin ylläpitäminen ja päivittäminen on vaivatonta, vaikka ohjelman käyttöikä olisi pitkä [5].

Koska työn tarkoituksena on särömittapaikan automatisointi, niin LabVIEW sopi suunnitteluun erinomaisesti. Arduinon ohjausta varten LabVIEW:iin voi ladata LabVIEW Makerhubin LINX -funktiokirjaston, joka on korvannut aiemmin käytössä olleen LIFA-funktiokirjaston. LINX on suunniteltu käytettäväksi yleisten sulautettujen alustojen, kuten Arduinon, chipKIT:n ja myRIO:n ohjauksessa ja kirjasto sisältää valmiiksi paljon tarvittavia toimintoja. Kuvassa 2 on LabVIEW:lla LINX-kirjaston avulla toteutettu Blink-ohjelma, joka kytkee Arduinon levyllä olevaa lediä sekunnin välein päälle ja pois. [6]



Kuva 2. LabVIEW'illa toteutettu blink-ohjelma.

LINX ei ole National Instrumentsin suunnittelema, vaikkakin National Instruments suosittelee sitä käytettäväksi sulautettujen alustojen ohjauksissa. LINX on Digilentin projekti ja perustuu avoimeen lähdekoodiin, eikä National Instruments takaa sen toimivuutta tai ylläpitoa. LINXiä kuitenkin päivitetään aktiivisesti, forumeilta saa sen kehittäjiltä apua ja sen

kehityksessä on mukana ammattilaisia, jotka ovat käyttäneet LabVIEW'tä jo hyvin pitkään. [6] [7]

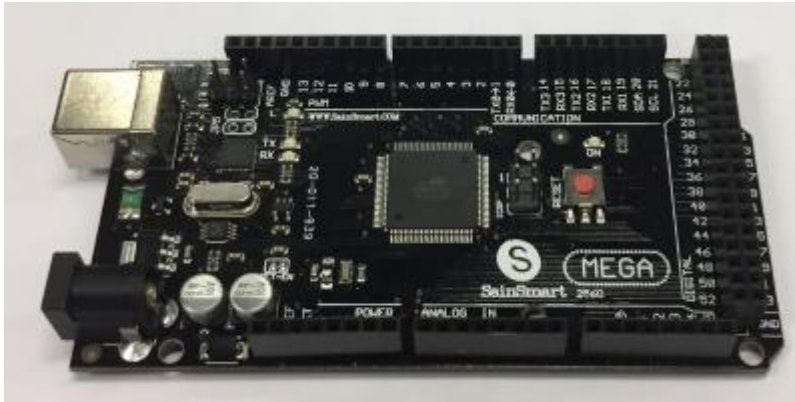
2.3 TestStand

TestStand on National Instrumentsin suunnittelema testienhallintaohjelmisto. Se on suunniteltu käytettäväksi automaattisissa testauksissa, joka myös tämän työn tarkoituksena oli. TestStandilla tehdyt mittausohjelmat, sekvenssit, koostuvat koodimoduleista, jotka voivat olla kirjoitettu eri ohjelmointikielillä. TestStand tukee kaikkia yleisimpiä ohjelmointikieliä, esimerkiksi LabVIEW:tä, Javaa, C++:aa, ja C#:a. Tämän vuoksi se on yleisesti käytössä testausympäristöissä. [8]

TestStand oli käytössä jo entuudestaan, joten uusien testausohjelmien implementointi ja testaus on nopeaa ja vaivatonta. Tärkeää oli huomata, että ajuri tuli suunnitella samalla tavalla kuin aiemmin käytössä olleet ajurit, jotta sen käyttöönottamiseksi ei tarvitse tehdä muutoksia aiempiin sekvensseihin tai moduuleihin.

2.4 Arduino

Arduino on avoimeen laitteistoon perustuva elektroniikka-alusta, jonka toiminta on pyritty luomaan moni- ja helppokäyttöiseksi. Arduino on noussut yhdeksi suosituimmista elektroniikka-alustoista ja sitä on ollut kehittämässä niin opiskelijoita, harrastajia kuin ammattilaisia. Arduinon käyttäjiä löytyy kaikkialta maailmasta ja käyttäjäyhteisö on aktiivinen, joten apua sen käyttämiseen ja ongelmatilanteisiin saa vaivattomasti. Arduinosta löytyy monta erilaista versiota, kuvassa 3 on Arduino Mega 2560. Eri käyttötarkoituksiin on mahdollista valita useamman eri Arduinon version väliltä. [9]



Kuva 3. Esimerkki Arduinosta, kuvassa Arduino Mega 2560.

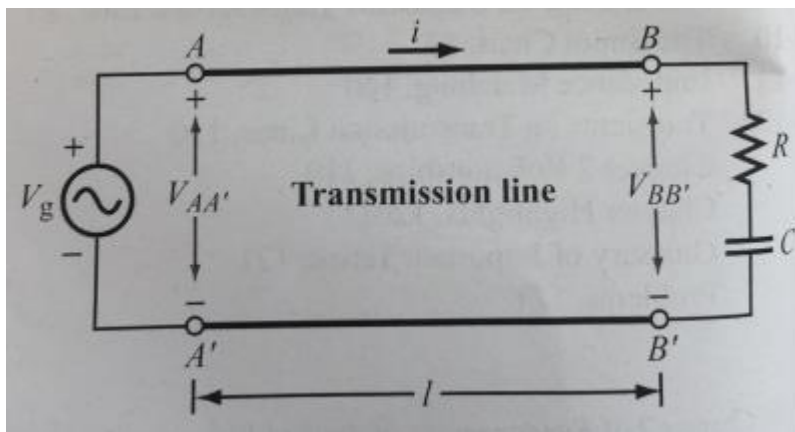
Arduinoa ohjelmoidaan sen omalla Arduino-ohjelmointikielellä, joka on kokoelma C ja C++ -kielillä toteutettuja funktioita. Arduinoa voi ohjelmoida myös normaaleilla C/C++ kielillä. Arduinossa on valmiina sen oma firmware, eli siihen kiinteästi asennettu ohjelmisto, joka vastaa sen perustoiminnoista. Kun käyttää LabVIEW Makerhub:n LINX-kirjastoa Arduinon ohjaamisessa, niin firmware on muutettava vastaamaan LINX-kirjaston vaatimia toimintoja. Ilman LINX-firmwarea Arduino ei osaa ottaa vastaan tietokoneelta tulevia, LINX-kirjaston avulla tehtyjä käskyjä. LINX-firmware on kuitenkin helposti asennettavissa Arduinoon LabVIEW:n kautta, LINX-kirjaston latauksen mukana tulee tarvittava firmware ja ohjeet sen asennukseen. [10]

3 RF-PIIRISUUNNITTELU JA SÄRÖT

Piiriä suunnitellessa täytyy osata ottaa huomioon monia asioita. Lähteen ja kuorman impedanssit eivät yleensä ole samoja, joten niiden välille voi tarvita sovituksen. Taajuuden kasvaessa komponentit alkavat käyttäytyä eri tavalla kuin DC:llä. Lisäksi signaalit alkavat ”hyppiä” siirtolinjalta toiselle sähkö- ja magneettikentän avulla. Suunnittelijan on tiedostettava monia eri asioita ja ymmärrystä on oltava monelta eri osa-alueelta, esimerkiksi sähkö- ja magneettikentistä sekä DC- ja AC-suunnittelusta. [1]

3.1 Siirtolinjat

Siirtolinjalla tarkoitetaan signaalin siirrossa kahden pisteen välistä yhteyttä, jota pitkin signaali voi edetä. Siirtolinja voi olla esimerkiksi koaksiaalikaapeli, punottu parikaapeli tai piirilevyllä oleva kuparifolio. Matalilla taajuuksilla siirtolinjan voi valita kohtuullisen vapaasti ilman suurempaa vaikutusta signaaliin. Korkeammilla taajuuksilla siirtolinjan valinta on kuitenkin otettava suunnittelussa huomioon. Kuvassa 4 on yksinkertainen kytkentä, jossa signaaligeneraattori on kytketty RC-piiriin. [11]

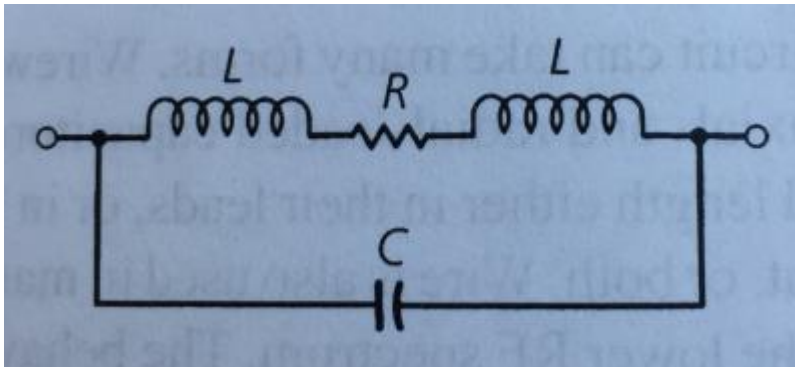


Kuva 4. Signaaligeneraattori kytkettynä siirtolinjalla RC-piiriin. [11]

Koska reaali maailmassa ei ole olemassa häviötöntä siirtolinjaa, niin $V_{AA'} \neq V_{BB'}$. Jos kuitenkin käytettävä siirtolinja on hyvänlaatuinen ja käytettävä taajuus on pieni (aallonpituus verrattuna siirtolinjan pituuteen on suuri), voidaan ajatella olevan $V_{AA'} \approx V_{BB'}$. Taajuuden kasvaessa ja aallonpituuden pienentyessä alkaa siirtolinjan pituus olla oleellinen tekijä suunnittelussa. Signaalin taajuuden kasvaessa siirtolinjan vaimennus kasvaa. [11], [1]

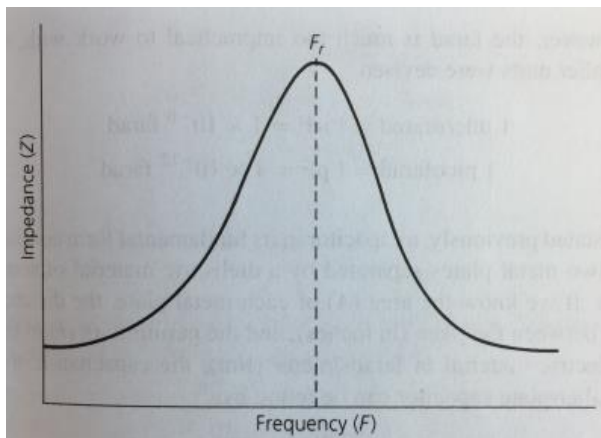
3.2 Vastus, kondensaattori ja kela

Vastus on yleisin komponentti, jota löytyy lähes kaikista piireistä. DC:llä sen voidaan ajatella omaavan vain yhden ominaisuuden, resistanssin. Kuitenkin AC:llä vastuksen muut ominaisuudet, induktanssi ja kapasitanssi, on otettava huomioon. Kuvassa 5 on vastuksen sijaiskytkentä radiotaajuuksilla, jossa L on vastuksen sähköä johtavan osan induktanssi, R on vastuksen resistanssi ja C on vastuksella oleva hajakapasitanssi. [1]



Kuva 5. Vastuksen sijaiskytkentä radiotaajuuksilla. [1]

Vastusten valmistustapoja on useita, eivätkä eri tavoilla valmistetut vastukset käyttäydy samalla tavalla korkeilla taajuuksilla. Esimerkiksi woundwire-vastuksen impedanssin käyttäytyminen taajuuden kasvaessa on kuvassa 6. [1]

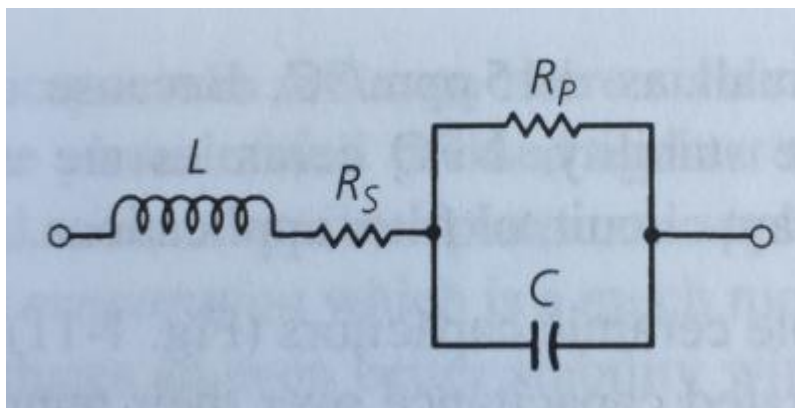


Kuva 6. Woundwire-vastuksen impedanssin käyttäytyminen taajuuden kasvaessa. [1]

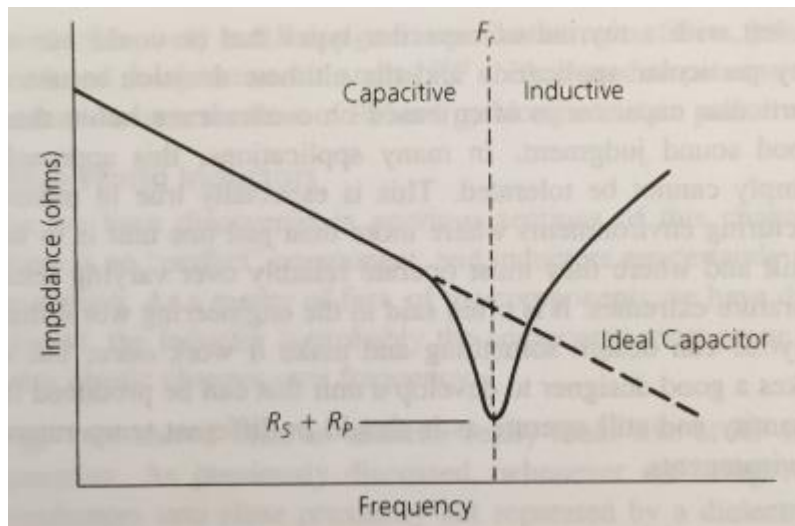
Koska woundwire-vastus on valmistettu kiertämällä johdinta jostain materiaalista valmistetun ytimen ympärille kelaksi, sen L on suuri ja saa sen käyttäytymään kelan tavoin taa-

juuden kasvaessa. Jollain tietyllä taajuudella F_r vastuksen L ja C resonoivat, synnyttäen impedanssiipiikin. Kun taajuus vielä kasvaa, alkaa impedanssi jälleen tippua, mutta vastus alkaa käyttäytyä kondensaattorin tavoin. [1]

Kondensaattoria käytetään usein RF-suunnittelussa, esimerkiksi suodattimissa, resonaattoreissa ja DC-katkoina. Kondensaattorin ominaisuudet määräytyvät suurimmilta osin sen eristeen vaikutuksesta. Kuvassa 7 on kondensaattorin sijaiskytkentä, jossa L on sen johtavien osien induktanssi, C on kapasitanssi, R_s on lämpöhäviö ja R_p on vuotovirran resistanssi. Kondensaattorin epäideaalisuuksien vaikutukset näkyvät kuvassa 8. [1]



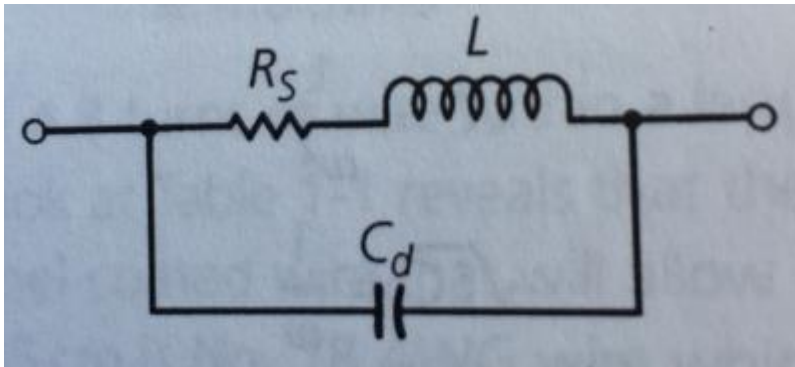
Kuva 7. Kondensaattorin sijaiskytkentä. [1]



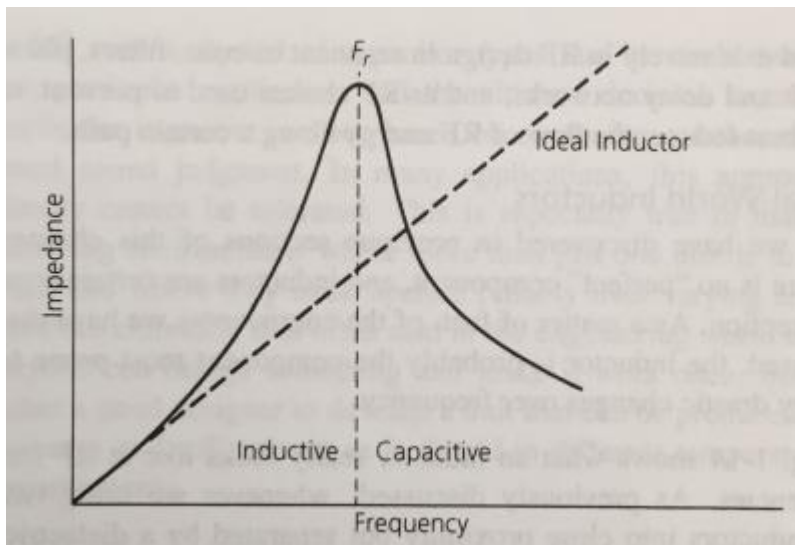
Kuva 8. Kondensaattorin impedanssin käyttäytyminen taajuuden kasvaessa. [1]

Alemmilla taajuuksilla impedanssi on kapasitiivista, mutta jollain taajuudella F_r kondensaattorin sarjassa olevat C ja L resonoivat, jota korkeammilla taajuuksilla kondensaattorin impedanssi on induktiivista. [1]

Kelaa käytetään RF-piireissä esimerkiksi suodattimissa, resonaattoreissa ja RF-kuristimena. Kuvassa 9 on kelan sijaiskytkentä, jossa L on kelan induktanssi, R_s kelan resistanssi ja C_d on kelan kierrosten välillä oleva kapasitanssi. Kelan epäideaalisuuksien vaikutukset on esitetty kuvassa 10. [1]



Kuva 9. Kelan sijaiskytkentä. [1]



Kuva 10. Kelan impedanssin käyttäytyminen taajuuden kasvaessa. [1]

Alhaisilla taajuuksilla kela muistuttaa ideaalista, mutta pian reaktanssi alkaa kasvaa nopeammin. Taajuudella F_r kelan rinnakkain kytketyt L ja C alkavat resonoida, ja korkeammille taajuuksille siirryttäessä kela alkaa muistuttaa kondensaattoria. [1]

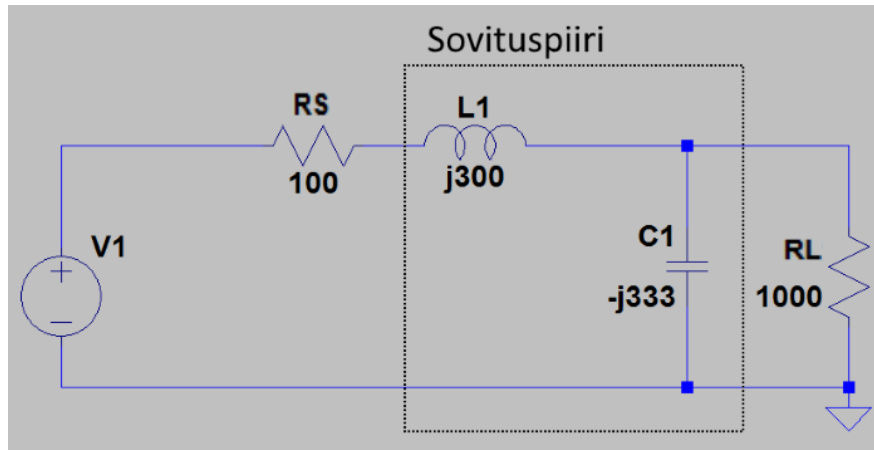
3.3 Sovituspiirit

Piirisuunnittelussa on yleistä, että lähteen impedanssi pitää sovittaa kuorman impedanssiin. Sovitus tarkoittaa, että kuorma muunnetaan ”näyttämään” impedanssiltaan samalta kuin lähde, jolloin tehosta saadaan siirrettyä mahdollisimman paljon kuormaan. Jos lähdettä ja kuormaa ei sovita, osa tehosta heijastuu takaisin lähteeseen ja pahimmillaan aiheuttaa siihen vauriota. Voi myös syntyä seisovia aaltoja, jotka tuottavat häviöitä. Mitä suurempi taajuus ja pidempiä siirtolinjat ovat, sitä suurempia häviöitä seisovat aallot aiheuttavat. [1]

AC-piirin suunnittelussa täytyy impedansseja sovitettaessa usein ottaa taajuus huomioon, mikäli sovitus halutaan tietylle taajuudelle. Taajuusriippuvaisia sovituspirejä on teoriassa äärettömän paljon, mutta useimmiten yksinkertainen sovituspiiri riittää lähes yhtä hyvän lopputuloksen aikaan saamiseen kuin monimutkaisempi sovituspiiri. Kolme yleisintä ja hyvin yksinkertaista sovituspiirimallia ovat L-sovituspiiri, Π -sovituspiiri ja T-sovituspiiri. L-sovituspiirissä on vain kaksi komponenttia, Π - ja T-sovituspireissä kolme. Π - ja T-sovituspiirit ovat käytännössä ekvivalentteja ja toimivat samalla tavalla, L-sovituspiiri eroaa niistä jonkin verran. [1]

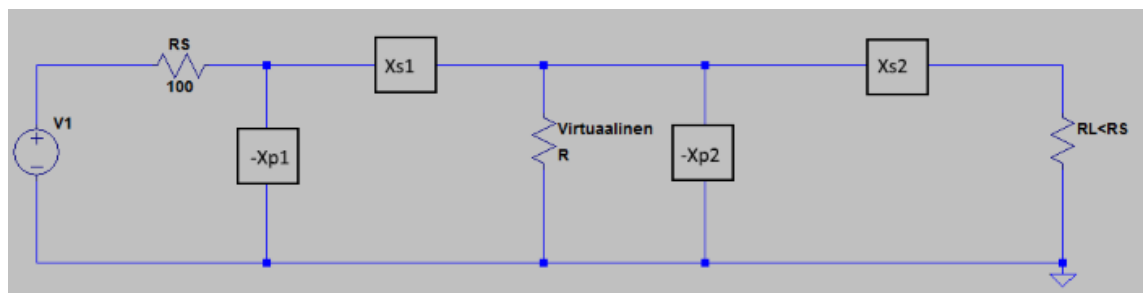
3.3.1 L-sovituspiiri

L-sovituspiiri on sovituspireistä yksinkertaisin, piirissä on vain kaksi komponenttia, jotka on sijoitettu sovitettavien impedanssien väliin L:n muotoon (kuva 11). Riippuen käytetyistä komponenteista, tulee sovituksesta joko yli- tai alipäästöinen. L-sovituspiirin tarkoituksena on sovittaa lähteen ja kuorman impedanssit lisäämällä komponentit impedanssiltaan suuremman rinnalle ja impedanssiltaan pienemmän kanssa sarjaa. Näin pyritään kuorman impedanssista tekemään lähteen impedanssin kompleksikonjugaatti. Kuvassa 11 olevassa esimerkissä lähde ja kuorma ovat puhtaasti resistiivisiä, mutta reaali maailmassa suunnittelussa näin on harvoin, yleensä lähde ja kuorma ovat kompleksisia. [1]

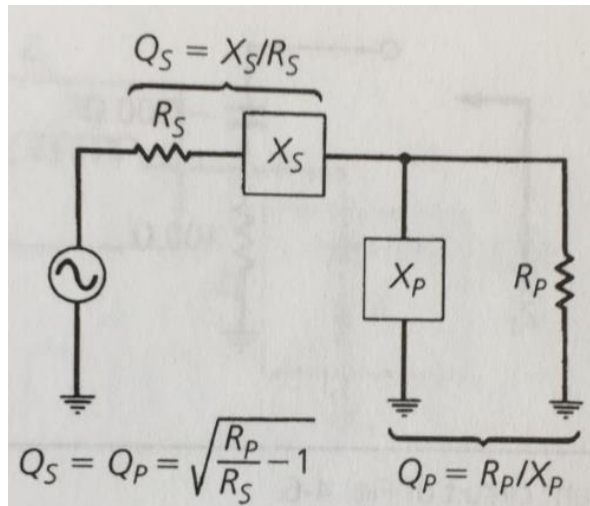


Kuva 11. Esimerkki alipäästöisestä L-sovituspiiristä.

L-sovituspiirin heikkoutena on se, että lähde ja kuorma määräävät piirin Q -arvon. Näin ollen sovituksen kaistanleveyttä ei pysty suunnittelemaan. Se voi tuottaa suunnittelussa ongelmia, varsinkin jos tarkoituksena on suunnitella kapeakaistaisella taajuusalueella toimiva sovituspiiri. Kapeakaistaiseen sovituspiiriin on L-sovituspiirin sijasta käytettävä esimerkiksi Π - tai T-sovituspiiriä. Jos taas on suunniteltava leveäkaistainen sovituspiiri, on mahdollista käyttää kahta tai useampaa sarjaan kytkettyä L-sovituspiiriä (kuva 12). Kuvasta 13 ilmenee L-sovituspiirin yhteenveto ja suunnitteluun tarvittavat laskukaavat. [1]



Kuva 12. Suuremman Q -arvon saamiseksi kaksi L-sovituspiiriä kytkettynä sarjaan.



Kuva 13. Yhteenveto L-sovituspiiristä. [1]

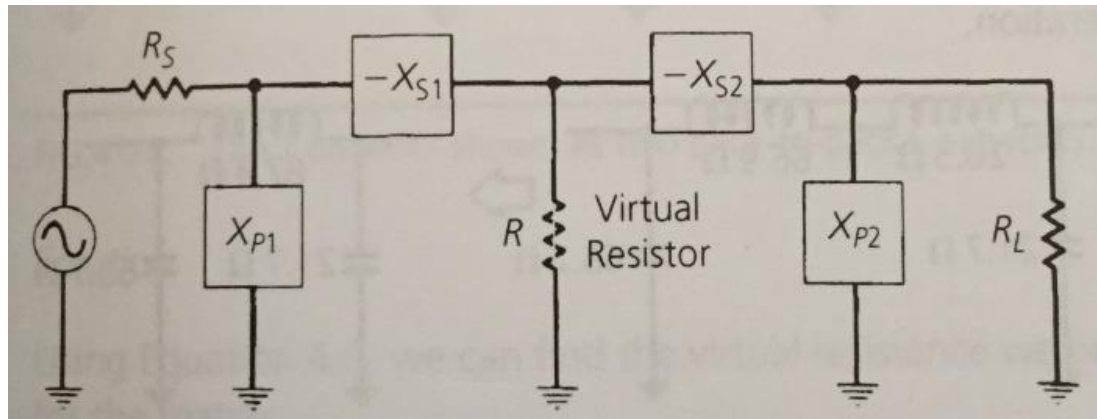
X_S ja X_P voivat olla reaktanssiltaan joko induktiivisia tai kapasitiivisia, mutta niiden on oltava vastakkaiset. Esimerkiksi jos X_S on kapasitiivinen, niin X_P :n on oltava induktiivinen. [1]

3.3.2 Π - ja T-sovituspiirit

Π -sovituspiirin suunnittelu vastaa lähes täysin L-sovituspiirin suunnittelua, koska suunnittelu on kahden L-sovituspiirin suunnittelu ja yhdistäminen (kuva 14). Kahden L-sovituspiirin avulla R_S ja R_L sovitetaan niiden välillä olevaan virtuaaliseen resistanssiin, joka on pienempi kuin kumpikaan niistä. Virtuaalinen resistanssi määräytyy yleensä halutun Q -arvon mukaan. Q -arvo voidaan laskea kaavalla [1]:

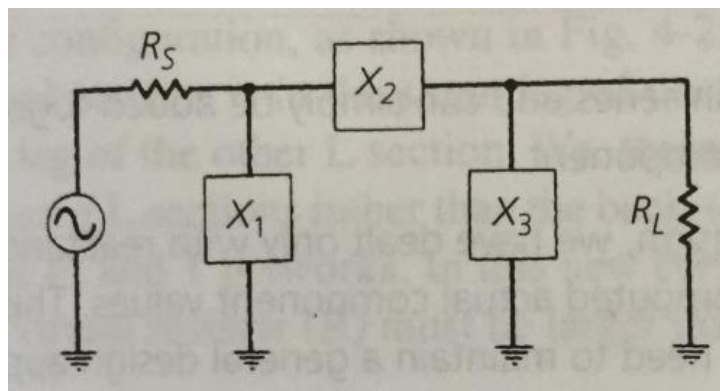
$$Q = \sqrt{\frac{R_H}{R} - 1}, \quad (3.1)$$

jossa R_H on sovitettavista impedansseista (R_S ja R_L) suurempi ja R on virtuaalinen resistanssi. Kaava ei ole aivan täsmällinen, mutta se on yleisesti hyväksytty kaava Π -sovituspiirin Q -arvon laskemiseen. Kun R on päätetty, kahden L-sovituspiirin laskeminen tapahtuu kuin edellä. [1]



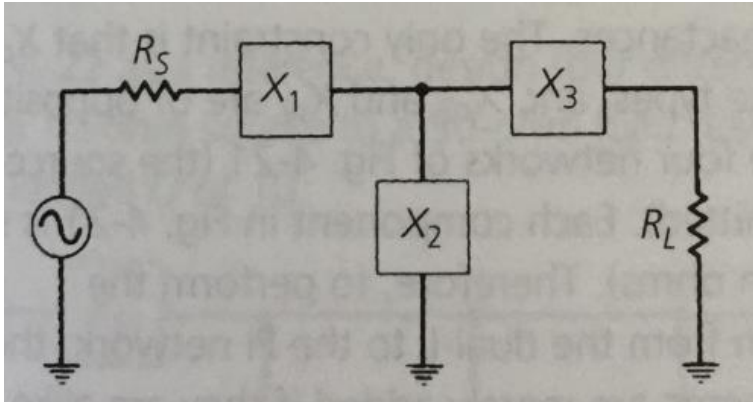
Kuva 14. Π -sovituspiiri esitettynä kahden L-sovituspiirin yhdistelmällä. [1]

Lopuksi sarjassa olevat komponentit X_{S1} ja X_{S2} voidaan yhdistää yhdeksi komponentiksi (kuva 15). Kuvassa 14 oleva miinusmerkki X_{S1} :n ja X_{S2} :n edessä on symbolinen, se merkitsee reaktanssin vastakkaisvaiheisuutta verrattuna rinnalle kytkettyihin X_{P1} ja X_{P2} :een. [1]



Kuva 15. Π -sovituspiiri, jossa kolme komponenttia. [1]

T-sovituspiirin suunnittelu vastaa Π -sovituspiirin suunnittelua suurilta osin. Erona on se, että virtuaalinen resistanssi on suurempi kuin kumpikaan sovitettavista impedansseista ja rinnan kytkettävät komponentit voidaan lopuksi yhdistää (kuva 16). [1]



Kuva 16. T-sovituspiiri, jossa kolme komponenttia. [1]

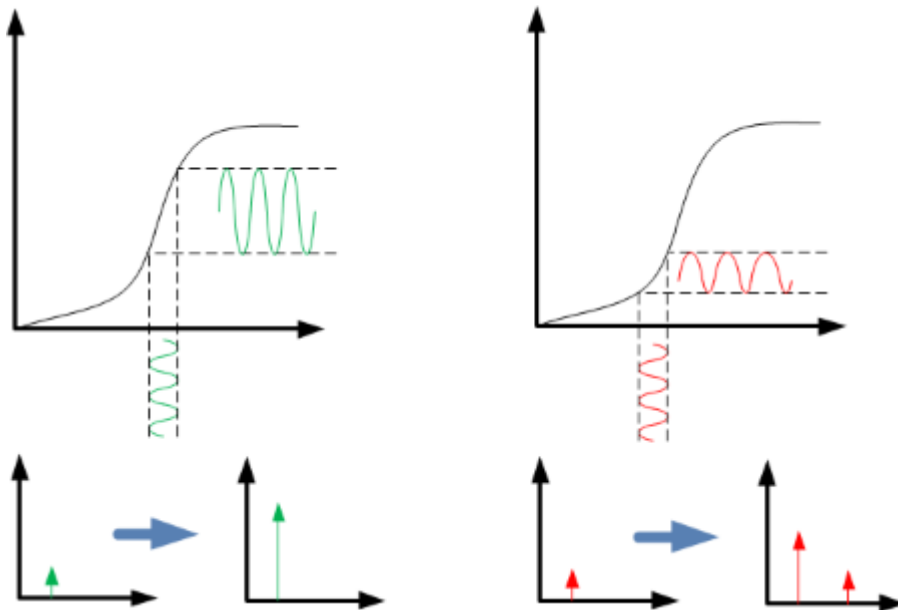
Piirin Q-arvo voidaan laskea kaavalla [1]:

$$Q = \sqrt{\frac{R}{R_{small}} - 1}, \quad (3.2)$$

jossa R on virtuaalinen resistanssi ja R_{small} on pienempi sovitettavista impedansseista. T-sovituspiiriä kannattaa käyttää, mikäli sovitettavat impedanssit ovat pieniä. [1]

3.4 Säröt

Eräitä kaapelitelevisiojärjestelmien laitteiden tärkeimmistä ominaisuuksista ovat särö-ominaisuudet. Särö tarkoittaa signaalin epäpuhtautta, jota syntyy erinäisistä syistä. Lineaarista säröä syntyy esimerkiksi passiivisissa komponenteissa tai siirtolinjoissa, jotka eivät ole täysin homogeenisia. Epälineaarista säröä puolestaan voi syntyä esimerkiksi epälineaarisista laitteista tai systeemeistä. Esimerkiksi jonkin vahvistimen ominaiskäyrä voi olla kuvassa 17 esitetyn näköinen, jolloin sillä on epälineaarinen ja lineaarinen toiminnallinen alue. [12]



Kuva 17. Esimerkki vahvistimen säröominaisuuksista.

Kuvassa vasemmalla vahvistimeen viedään signaali sen lineaarisella toiminta-alueella ja ulos tulee vahvistunut signaali, joka on kuitenkin saman muotoinen. Lisäksi signaalin spektrissä näkyvä signaali on vahvistunut. Jos signaali tuodaan vahvistimeen sen epälineaarilla toiminta-alueella (kuvassa oikea), niin syntyy signaalin spektriin uusia komponentteja, epälineaarisia säröjä, ja signaalin muoto vääristyy. [13]

Kaapelitelevisiojärjestelmien kanta-aallot ovat hyvin lähellä toisiaan, jonka johdosta säröjä syntyy hyvin lähelle kanta-aaltoja vaimentamattomana. Erityisesti usean kanta-aallon aiheuttamat säröt CTB ja CSO voivat tuottaa ongelmia kaapelitelevisiojärjestelmissä. [12]

Toisen kertaluokan epälineaarista säröä (CSO) muodostuu kahden kanta-aallon f_1 ja f_2 kokiessa epälineaarista säröytymistä ja särön taajuus on $f_1 \pm f_2$. Esimerkiksi käytettäessä PAL-B systeemiä kaapelitelevisiojärjestelmissä CSO aina vähintään 0,25 MHz:n päässä kanta-aallosta. Esimerkiksi $294,25 \text{ MHz} - 238,25 \text{ MHz} = 56 \text{ MHz}$, joka on 0,25 MHz:n etäisyydellä kuvankanta-aallosta taajuudella 55,75 MHz. [12]

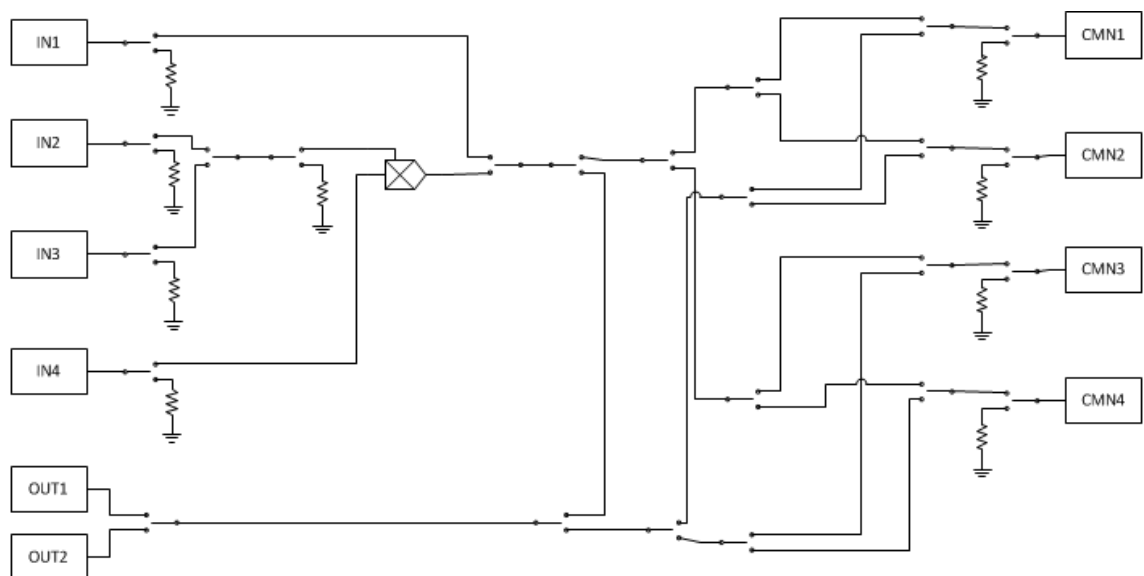
Kolmannen kertaluokan epälineaarista säröä (CTB) muodostuu puolestaan kolmen kanta-aallon f_1, f_2 ja f_3 kokiessa epälineaarista säröytymistä. Muodostuneista säröistä osa ei muodostu lainkaan mitattavalle alueelle, mutta säröt $f_1 + f_2 - f_3$ ja $f_1 - f_2 + f_3$ voivat osua samalle taajuudelle kuin kanta-aalto. [12]

4 TYÖN SUUNNITTELU

Työn suunnittelu jaettiin pienempiin osiin kokonaiskuvan muodostamisen ja työn aikataulutuksen helpottamiseksi. Toimeksiantoyritystä informoitiin työn etenemisestä viikottain järjestetyissä palavereissa, joissa työn etenemistä kuvattiin kanban-taululla. Käytäntö on osa Leanin periaatteita, visuaalisuus mahdollistaa helpomman arvioinnin työn osien kestosta ja koosta [3]. Tärkeimmät suunnittelun osat olivat lohkokaavio, RF-kytkinten ohjaukset, komponenttien valinnat ja piirin suunnittelu.

4.1 RF-lohkokaavion suunnittelu

RF-lohkokaavioon sisällytettiin vain RF-siirtolinjojen osat, eli liittimet, RF-kytkimet, summain ja siirtolinjat. Kytkinten ohjauksia tai käyttö sähköä ei huomioitu tässä vaiheessa, koska ne tulivat joka tapauksessa piirilevyn eri tasolle. Lohkokaaviosta selvisi nopeasti, että joitain RF-siirtolinjoja menee levyllä ristiin. Lohkokaavio on esitetty kuvassa 18.



Kuva 18. Moniveräjän RF-lohkokaavio.

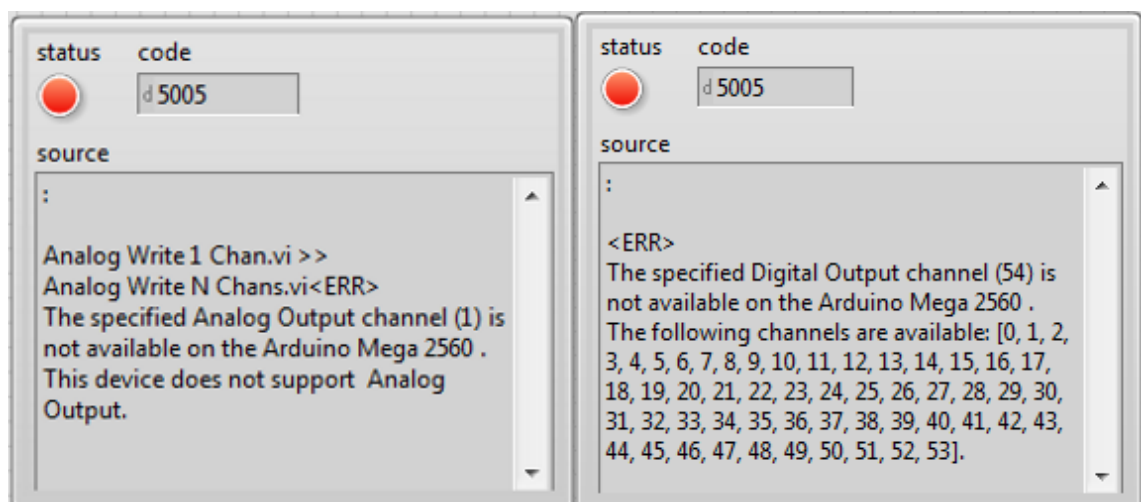
Kaikki IN-portit tuli olla mahdollista päättää, jotta niissä ei tapahtuisi liian suurta signaalin heijastumista kun ne eivät ole käytössä. Samoin kaikki Common-portit (CMN), joihin DUT kytketään, tuli pystyä päättämään. OUT-portteja, joihin kytketään spektrianalysaattori ja tehomittari, ei tarvinnut pystyä päättämään. Lisäksi summainmelle IN2 ja IN3:lta tuleva

linja tuli pystyä päättämään. IN- ja OUT-portit tuli myös pystyä yhdistämään toisiinsa suoraan.

RF-lohkokaaviosta oli mahdollista ennustaa mahdollisia ongelmakohtia piirisuunnittelussa. Risteävät RF-linjat olivat yksi ongelma, ne oli mahdollista piirtää eri kerroksiin piirilevyllä, tai ladonnan jälkeen juottaa linjan tilalle koaksiaalikaapeli. Summaimelle portista IN4 tulevaa linjaa ei voi päättää kytkimellä, sillä se oli alun perin tarkoitus päättää signaaligeneraattorilla. Tarkoituksena oli pitää RF-kytkimien määrä mahdollisimman vähäisenä signaalireiteillä.

4.2 RF-kytkinten ohjausten suunnittelu ja Arduinon valinta

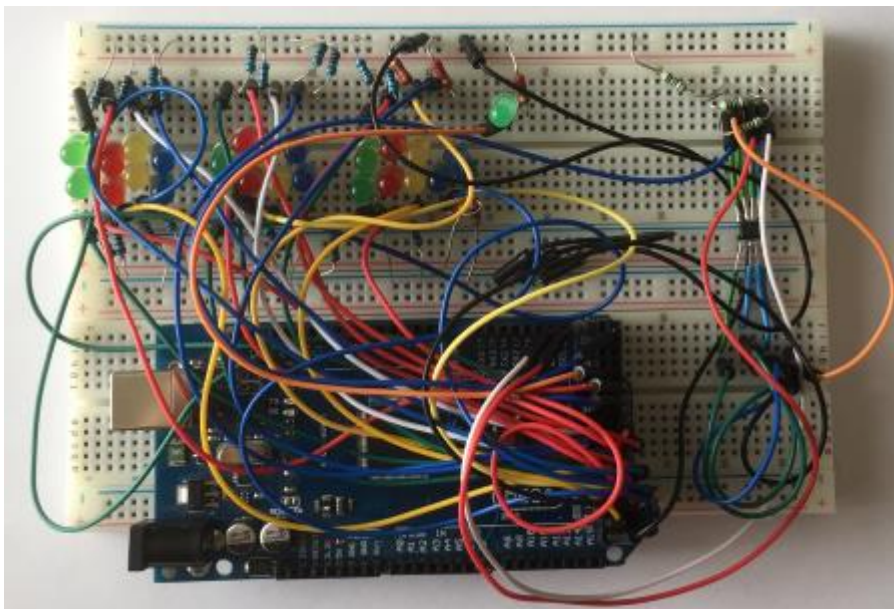
Lohkokaavion perusteella ohjaussignaaleja 24 RF-kytkimelle tarvittiin vähintään 18. LINX-kirjaston käyttö rajoittaa pinnejä, joita Arduinosta voi käyttää, sillä LINX-kirjaston avulla analogisia pinnejä ei voi käyttää digitaalisina I/O-pinneinä, joita työn RF-kytkimien ohjauksissa tarvitaan. Kuvassa 19 on esitetty LabVIEW:n palauttavat error-viestit yritettäessä käyttää Arduino Mega 2560:n analogisia pinnejä A0 ja A1 lähtöinä. LINX:iin ei ole vielä implementoitu Arduinoa varten analogisia lähtöjä. LINX:ä päätettiin kuitenkin käyttää, sillä omien kirjastojen tekeminen Arduinon ohjaukseen veisi paljon aikaa. Esimerkiksi Arduino Mega 2560:ssä on ilman analogisia pinnejä käytössä vielä 38 digitaalista I/O-pinniä, jotka riittävät mainiosti RF-kytkimien ohjauksiin. [14]



Kuva 19. Error-viestit yritettäessä käyttää Arduinon analogista pinniä LINX-kirjaston avulla.

Käyttöön valittiin Arduino Mega 2560 ja siitä valittiin I/O-pinnit sillä tavalla, että piiri-suunnittelu ja koodin tekeminen olisivat mahdollisimman helppoja. Koska I/O-pinnejä oli käytössä runsaasti, päätettiin jokaista 24 kytkintä ohjata omalla pinnillä koodin yksinkertaistamiseksi. Lisäksi kävi ilmi, että I2C-väylän pinnejä D20 ja D21 tarvittaisiin ulkoisen EEPROMin hallintaan, sekä vielä yhtä I/O-pinniä releen ohjauksessa. Pinneiksi valittiin D29 — D53, yhteensä 25 pinniä.

Arduinoa päätettiin vielä testata toimivuuden varmistamiseksi. Testausta varten kytkentälevylle rakennettiin kuvassa 20 oleva yksinkertainen kytkentä, jossa haluttuihin pinneihin on kytketty led, joka on kytketty vastuksen kautta maihin. Samaa kytkentää käytettiin myöhemmässä vaiheessa, kun piirilevy ei ollut vielä valmistunut ja ladottu, mutta ajuria täytyi pystyä testaamaan.



Kuva 20. Arduino Mega 2560:n testauksessa käytetty kytkentä.

Mittauksissa todettiin, että kaikki pinnit toimivat halutulla tavalla. Mega 2560:n 3,3 V:n pinnin teho saadaan regulaattorilta ja maksimivirta 3,3 V:n pinnistä on 50 mA, joka riittää RF-kytkimien käyttö sähköihin mainiosti [14].

4.3 RF-kytkinten ohjausten ja käyttöjännitteiden suunnittelu

RF-kytkinten ohjaukset ja käyttöjännitteet pyrittiin suunnittelemaan niin, etteivät siirtolinjat ristenneet missään kohtaa. Silloin siirtolinjat oli mahdollista tehdä piirilevyssä yh-

delle kerrokselle. Koska kyseisissä siirtolinjoissa ei kulje suuritaajuuksisia signaaleja, vaan tasajännitettä, oli ohjauksien lohkokaaavion suunnittelussa paljon enemmän vapauksia kuin RF-lohkokaaavion suunnittelussa. Tasajännite ei hyppää siirtolinjalta toiselle, vaikka siirtolinjat kulkisivat levyllä samansuuntaisesti pitkänkin matkan [1].

4.4 Komponenttien valinnat

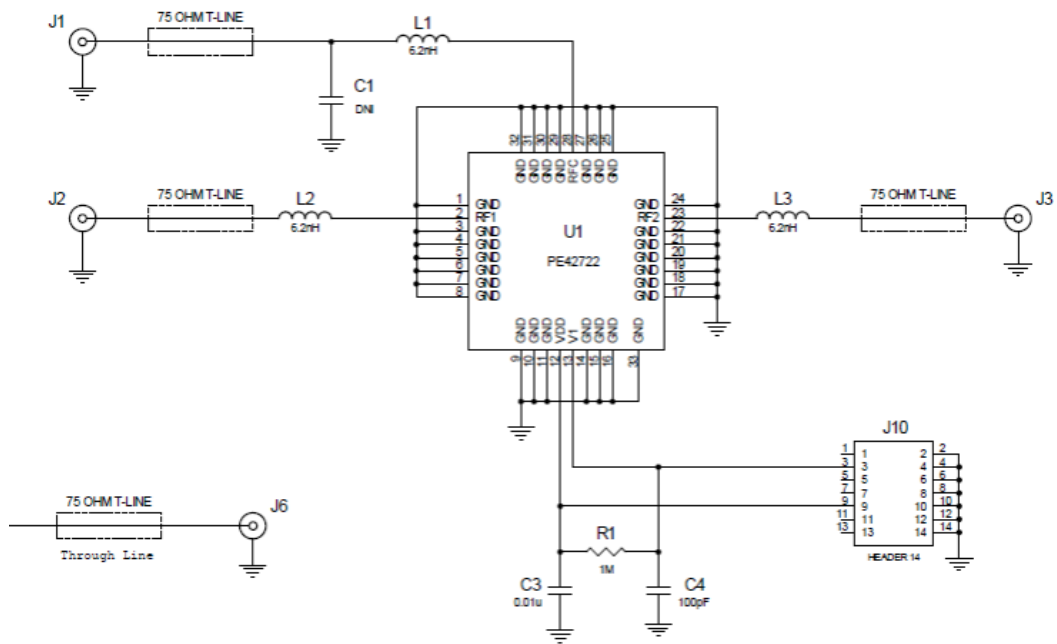
Komponenttien valinnoissa yritettiin käyttää mahdollisimman paljon jo entuudestaan tuotannossa käytössä olleita komponentteja, jotta tilaaminen oli helppoa ja piiriä suunniteltaessa komponentit löytyivät jo valmiiksi piirisuunnitteluohjelman kirjastosta. Moniveräjään ei tarvinnut kovin laajaa kirjoa komponentteja, joka osaltaan helpotti suunnittelua. Piirissä päätettiin käyttää samoja vastuksia, kondensaattoreita ja keloja kuin mitä tuotannossa oli käytössä entuudestaan. Niiden valintaan ei käytetty aikaa, sillä niiden ominaisresonanssitaajuuksien tiedettiin olevan suurempia, kuin moniveräjän taajuusalueen yläraja.

4.4.1 RF-kytkin

RF-kytkimeksi valittiin DOCSIS 3.1 Standardin vaatimusten mukaisesti suunniteltu Peregrine Semiconductorsin valmistama PE42722, joka toimii taajuusalueella 5 – 1794 MHz ja on suunniteltu säröominaisuuksiltaan hyväksi. Lisäksi sen isolaatio on suurempi kuin 26 dB koko taajuusalueella. Siinä on myös sisäisesti toteutettu ESD suoja 1,5 kV:iin asti (Human Body Model). [15]

PE42722:ssa on käytetty Peregrine Semiconductorsin patentoimaa UltraCMOS-teknologiaa, joka tarjoaa paremman toimintavarmuuden pienellä teholla ja RF-taajuuksilla. Sillä saadaan myös valmistettua komponentteja, joilla on pienempi heijastusvaimennus, parempi lineaarisuus ja paremmat säröominaisuudet kuin esimerkiksi galliumarsenidiin perustuvilla komponenteilla. Lisäksi UltraCMOS tekniikalla valmistetut IC-piirit ovat immuuneja latch up -ilmiölle, joka voisi pahimmillaan tuhota IC-piirin. [16], [17]

PE42722:n tavanomainen kytkentä saatiin kysymällä Peregrine Semiconductorsin tekniestä tuesta ja se on esitetty kuvassa 21. Kelat ovat tarkoitettu kytkimen ja siirtolinjan sovitukseen, muut komponentit ovat valinnaisia. Ne on tarkoitettu mahdollistamaan kytkimen ohjaus hyppykaapelilla ilman ohjauslogiikkaa. [18]



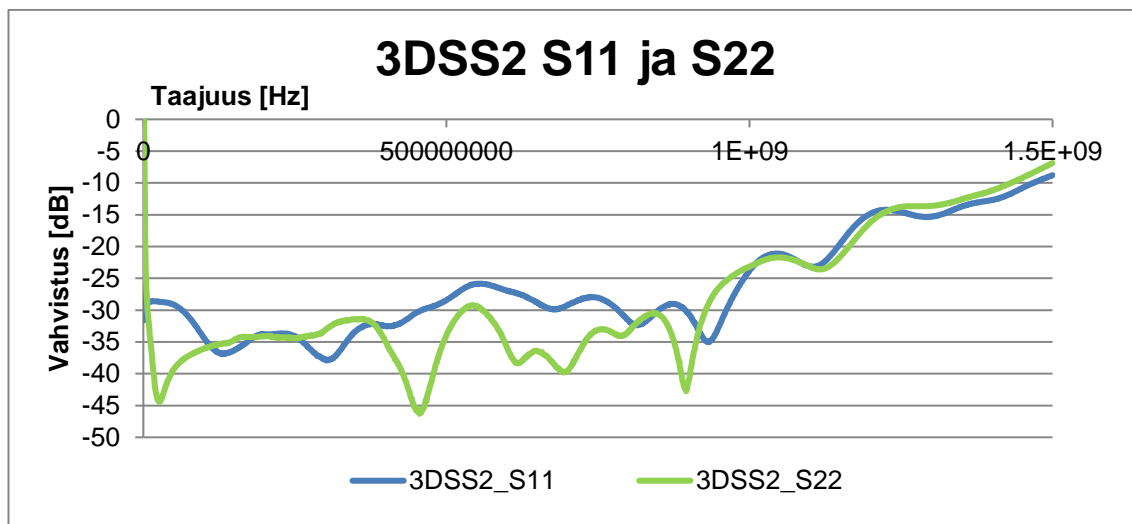
Kuva 21. PE42722:n tavanomainen kytkentä. [18]

Tavanomaisen kytkennän perusteella osattiin valita sopivat komponentit, joita RF-kytkin tarvitsee piirilevylle.

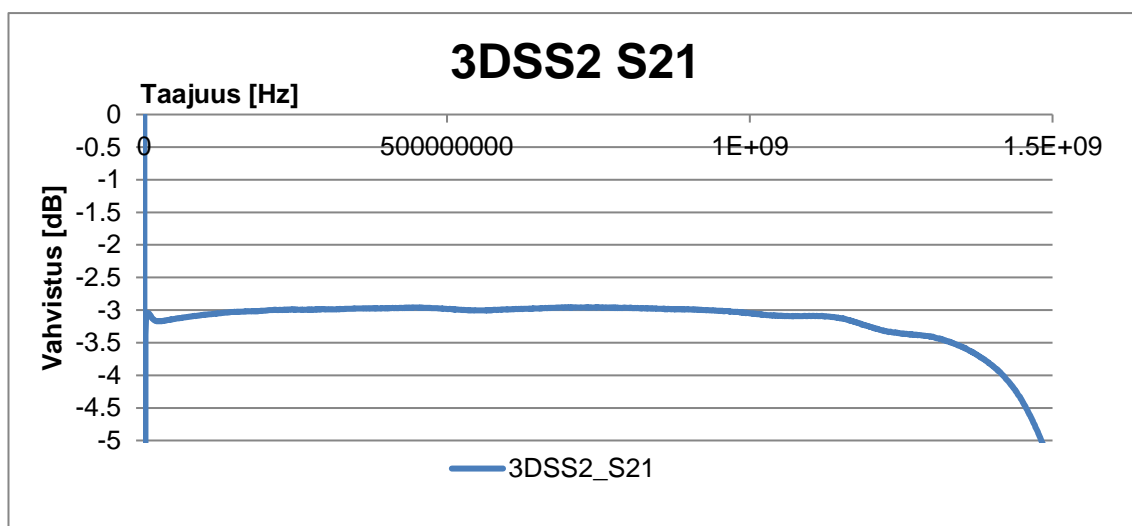
4.4.2 Summain

Summaimeksi valittiin 3DSS2, joka on suunniteltu DOCSIS 3.1 standardin vaatimusten mukaisesti ja toimii taajuusalueella 5 – 1200 MHz. Sen isolatio on suurempi kuin 23 dB koko taajuusalueella ja se on helppo kiinnittää ja kytkeä piirilevylle. [19]

Summainen toiminta haluttiin varmistaa tilanteessa, jossa vain toiseen input-porttiin tulee signaali, kun taas toiseen input-porttiin tulee 75 Ω :n pääte. Mittaukset suoritettiin Agilent E5061A piirianalysoitsijalla, jonka tehona käytettiin -5 dBm:ä. Kuviossa 1 on mitattu HV ja kuviossa 2 mitattu läpäisy.



Kuvio 1. Summainen 3DSS2 heijastusvaimennus.



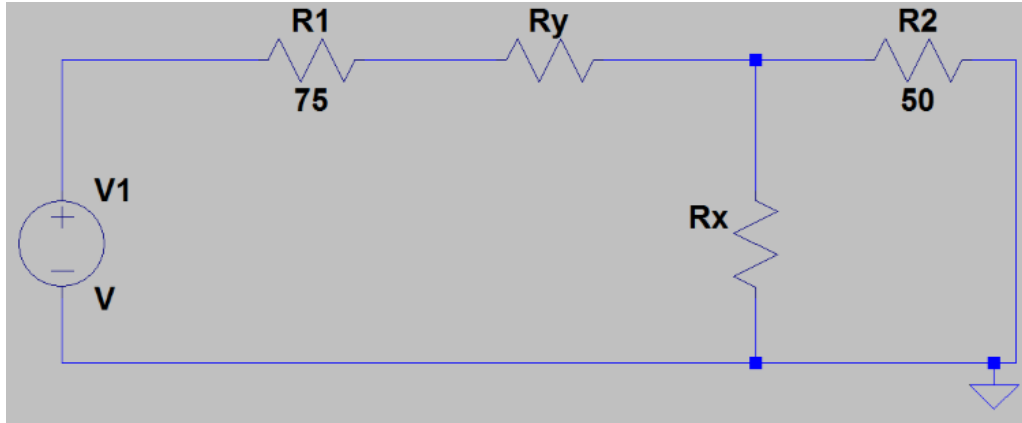
Kuvio 2. Summainen 3DSS2 läpäisy.

Mittauksissa todettiin, että summain toimi halutulla tavalla; datalehdessä ilmoitettu maksimivaimennus 4,6 dB:ä ja minimi HV 14,5 dB:ä taajuudella 1 200 MHz toteutuivat. Mittausten perusteella summain vaikutti toimivalta ratkaisulta käytettäväksi piirissä.

4.4.3 Sovituspiiri

Moniveräjän tuli pystyä reitittämään 50 ja 75 Ω :n impedanssia käyttäviä siirtolinjoja, joten 50 Ω :n siirtolinjat tuli sovittaa 75 Ω :iin. Sovituksessa harkittiin ensin käytettävän valmista

sovitinta, mutta lopulta levyllä päätettiin suunnitella sovituspiiri. Sovituspiiri on kohtuullisen helppoa suunnitella ja on halpa, kun taas valmis sovitin maksaa siihen verrattuna moninkertaisesti. Koska sovitusta tarvitaan suurelle taajuusalueelle, eikä vaimennus tuota ongelmia, päätettiin käyttää resistiivistä sovitusta, josta tulee helposti laajakaistainen [21]. Kuvassa 22 on esitetty sovituspiirin laskemiseen tarvittavat komponentit (siirtolinjoja edustavat kuvassa vastukset).



Kuva 22. Laskettavan sovituspiirin komponentit.

Seuraavasta yhtälöparista saatiin laskettua sopivat vastusarvot:

$$\begin{cases} 50 \, \Omega = \left(\frac{1}{R_X} + \frac{1}{R_Y + 75 \, \Omega} \right)^{-1} \\ 75 \, \Omega = R_Y + \left(\frac{1}{50 \, \Omega} + \frac{1}{R_X} \right)^{-1} \end{cases} \quad (4.1)$$

Ratkaisemalla yhtälöparista R_X ja R_Y , saatiin arvoiksi $R_X=86,6 \, \Omega$ ja $R_Y=43,3 \, \Omega$. Täsmälleen samoja vastusten arvoja ei kuitenkaan löydy E24-sarjasta, joten arvoiksi valittiin $R_Y=43 \, \Omega$ ja $R_X=82 \, \Omega$ [20]. Vaimennuksen voi laskea kaavalla 4.2 [21]. Täsmällisillä arvoilla lasketun vaimennuksen arvo ilmenee kaavasta 4.3, käytetyillä arvoilla laskettu vaimennuksen arvo taas kaavasta 4.4.

$$K_p = 10 \log \frac{(R_Y + R_1 + R_X)(R_Y + R_1)}{R_X * R_1} \quad (4.2)$$

$$K_p = 10 \log \frac{(43,3 \, \Omega + 75 \, \Omega + 86,6 \, \Omega)(43,3 \, \Omega + 75 \, \Omega)}{(86,6 \, \Omega) * (75 \, \Omega)} \approx 5,7 \, \text{dB} \quad (4.3)$$

$$K_p = 10 \log \frac{(43 \, \Omega + 75 \, \Omega + 82 \, \Omega)(43 \, \Omega + 75 \, \Omega)}{(82 \, \Omega) * (75 \, \Omega)} \approx 5,8 \, \text{dB} \quad (4.4)$$

Vaimennus ei tässä tapauksessa ole liian suuri, tärkeintä on sovituksen laajakaistaisuus.

5 TYÖN TOTEUTUS

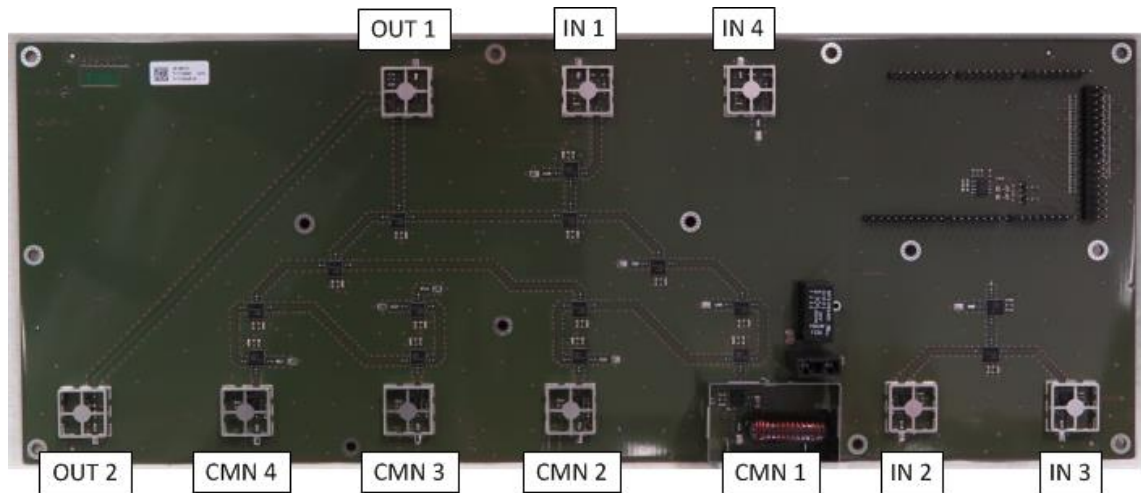
Huolellisen suunnittelun jälkeen tuli työn toteutus. Toteutus aloitettiin suunnittelemalla moniveräjän piirilevy, jonka jälkeen suunniteltiin moniveräjälle ajuri. Kun piirilevy oli valmistunut ja ajuri kunnossa, oli testausten ja parantelujen vuoro. Testauksiin kuului moniveräjän reittien HV:t ja läpäisy, sekä säröominaisuudet. Lopuksi oli vuorossa moniveräjän koteloinnin vaikutus mittaustuloksiin.

5.1 Piirin suunnittelu ja piirtäminen

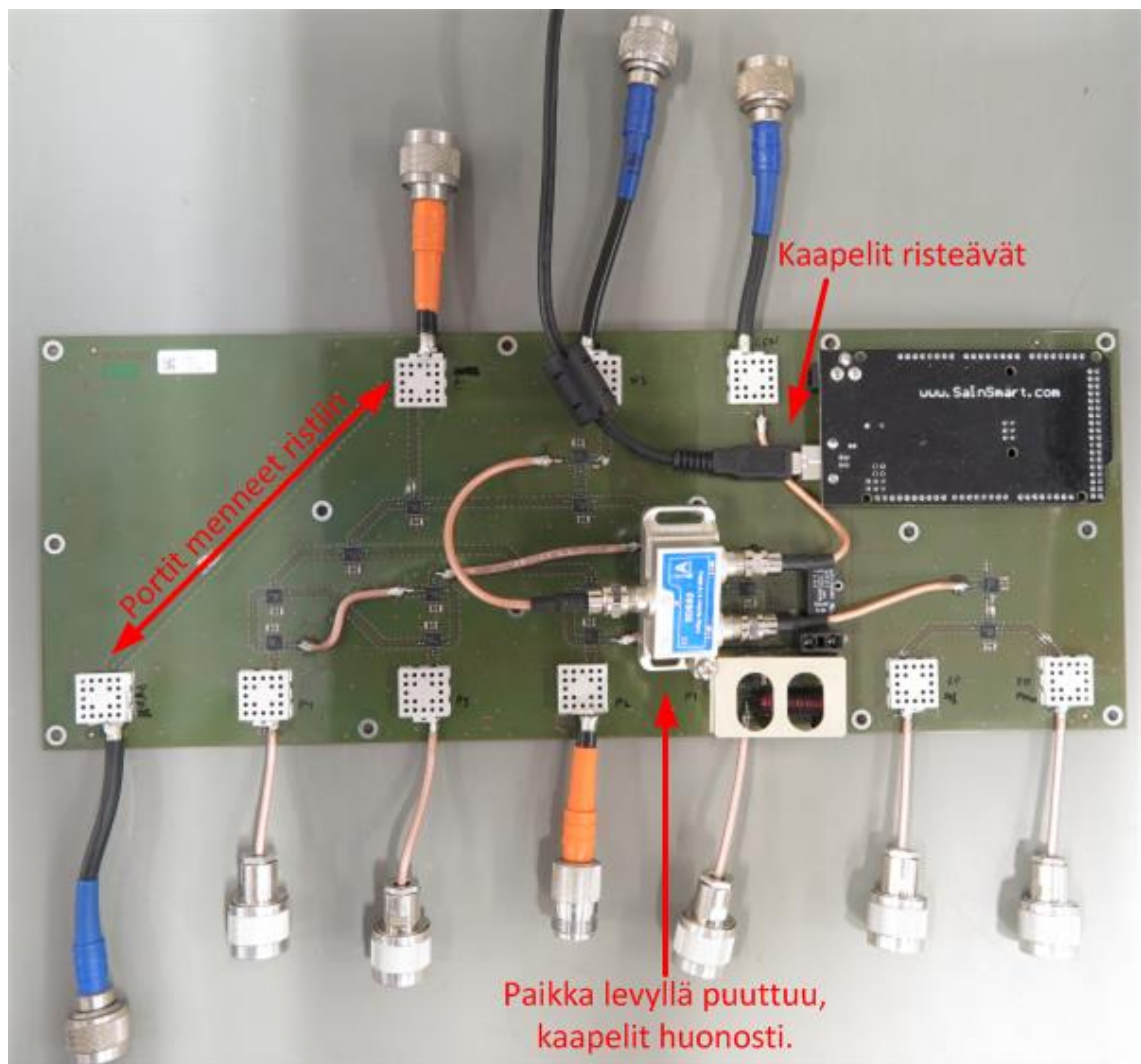
Piirin kaavion piirtäminen suoritettiin Mentor Graphicsin Board architect -piirisuunnitteluohjelmalla. Piiri on suurikokoinen, mutta sen kaavio oli helppo suunnitella, sillä esimerkiksi RF-kytkimien kytkennät ovat samanlaisia ja yhden piirrettyä sen sai kopioitua käytettäväksi muualla.

Piirin tulot ja lähdöt päätettiin tuoda levyille koaksiaalikaapelilla, joiden keskijohdin juotetaan suoraan levyllä olevalle siirtolinjalle. Se saattaa kuitenkin altistaa ylikuulumiselle, joten kaikki tulot ja lähdöt päätettiin suojata häiriösuojakotelolla. RF-kytkinten sovitukseen tarvittavia 6,2 nH:n keloja ei E24-sarjasta löydy, joten piirissä käytettiin 5,6 nH:n keloja [20].

Piirin layoutin piirsi piirilevysuunnittelija, joka hahmotteli sen valmiin suunnitelman mukaisesti. Piiri oli liian monimutkainen simuloitavaksi, joten sen toiminta päätettiin tarkastaa prototyyppillä, jonka mittaustulosten perusteella piiriä muutettiin vaatimusten saavuttamisen vaatimalla tavalla. Kuvassa 23 on ladottu piirilevy, johon on merkattu sen porttien nimet. Valmiiksi tehty piirilevy näkyy kuvassa 24, kuvaan on merkattu punaisella ilmenneitä ongelmakohtia.



Kuva 23. Ladottu monivieräjän piirilevy.



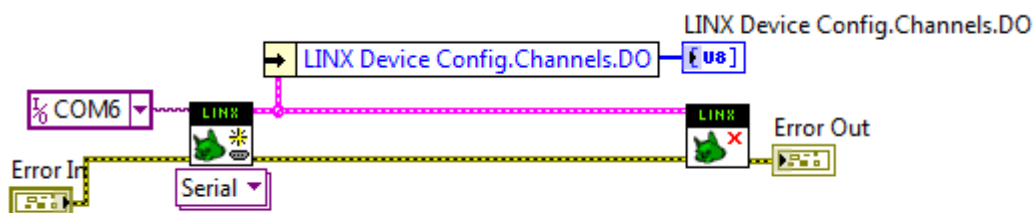
Kuva 24. Monivieräjän prototyyppi.

Piirilevyssä ilmeni heti aluksi joitain virheitä, jotka olisi korjattava seuraavaan versioon. Summainen 3DSS2 paikka piirilevyllä oli liian pieni, jolloin se oli kiinnitettävä piirilevyllä korotustornin avulla. Lisäksi IN 4 -portin koaksiaalikaapeli risteää Arduinon USB-kaapelin kanssa, jolloin Arduinoa ei voi painaa pohjaan asti. Myös 3DSS2:n lähdössä olevan koaksiaalikaapelin juotosasento oli epätoivottu, jonka johdosta siihen joutui juottamaan pitkän koaksiaalikaapelin. Lisäksi OUT1 ja OUT2 olivat suunnitelmassa menneet ristiin, jolloin piirilevyn samalla puolella oli 50 ja 75 Ω :n portteja, joka ei ollut toivottua.

5.2 Arduinon ohjaus tietokoneella

Ajuri tuli toteuttaa mahdollisimman paljon samalla tavalla, kuin jo entuudestaan käytössä olleet ajurit. Aiemmin oli luotu tietty formaatti, jota tuli noudattaa, jotta ajurin käyttöönotto on mahdollista. Ajurin VI tuli siis suunnitella siten, että se osaa käsitellä käskyt ja palauttaa tulokset oikeassa, jo aiemmin määritetyssä muodossa. Koska ajuria suunnitelmassa piirilevy ei ollut vielä valmistunut, käytettiin sen tilalla aiemmin esiteltyä testikytkentää (kuva 20).

Ajurin haluttiin toimivan myös käytettäessä toista samankaltaista laitetta, joka käyttää kuitenkin Arduino Nanoa. Nanosta oli käytetty ohjauksiin myös analogisia pinnejä, joten oli löydettävä keino käyttää niitä LINX-kirjaston avulla. Tutkimalla LINX-kirjaston toimintaa ilmeni, että käytössä olevat digitaaliset I/O-pinnit ovat kovakoodattu Arduinon LINX-firmwareen. Kuvassa 25 näkyy LabVIEW:ssä LINX-resource-klusterissa olevan LINX Device Config.Channels.DO -taulun hakeminen, josta löytyvät käytössä olevat digitaaliset output-pinnit, jotka ohjelma hakee Arduinolta.

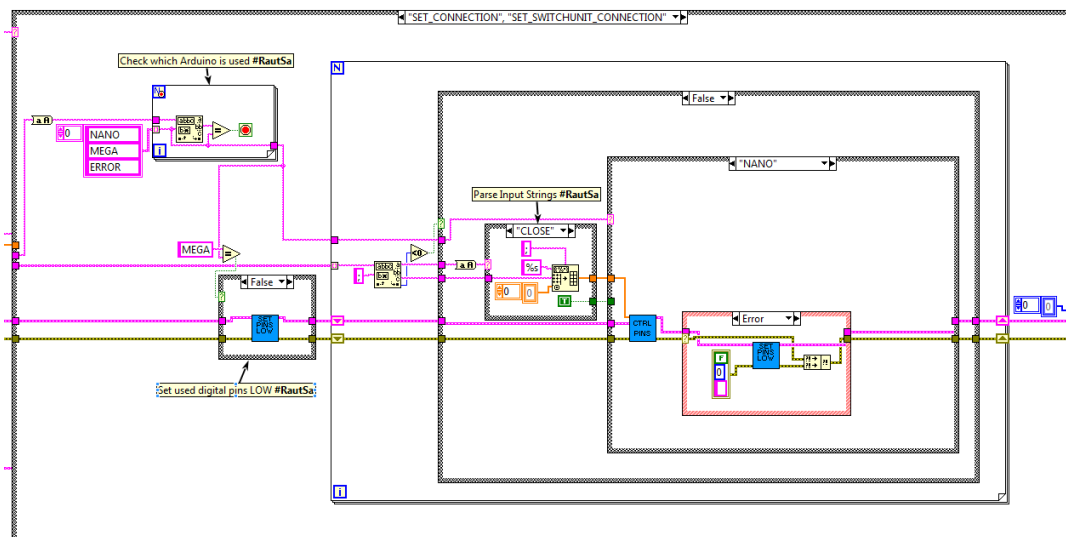


Kuva 25. Käytössä olevien digitaalisten lähtöjen taulun hakeminen LINX Resource -klusterista.

Muutoksen olisi voinut tehdä lisäämällä LabVIEW:n koodissa tarvittavien pinnien numerot kuvassa 25 näkyvän klusterin LINX Device Config.Channels.DO -tauluun. Muutos päätettiin kuitenkin tehdä LINX:n Arduinon firmwareen. Muutoksen Arduinon firmwareen toteutti testaussuunnittelija.

5.2.1 RF-kytkinten ohjaus

RF-kytkinten ohjausta varten oli ajuriin luotava toiminnallisuus, joka osaa valita oikean käytössä olevan Arduinon ja reitittää sen mukaan siirtolinjat. Kuvassa 26 näkyy kyseinen tapaus.



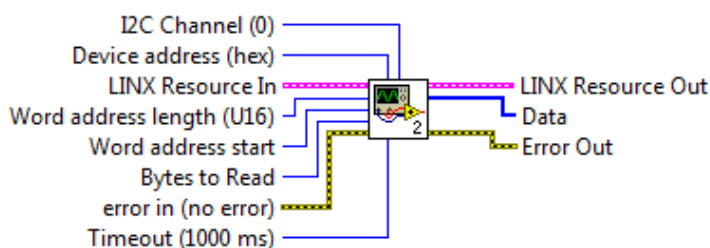
Kuva 26. Ohjelman osa, jossa tarkastetaan Arduinon malli ja toteutetaan haluttu kytkentä.

Piirilevy ja ajuri pyrittiin suunnittelemaan siten, että asettamalla aluksi kaikki Arduinon pinnit LOW-tilaan päätetään kaikki portit samalla. Näin vältetään tilanne, jossa jonkin lähteen signaali pääsee häiritsemään mittauksia. Piirilevyn ensimmäisessä versiossa oli erehdyksessä IN-porttien päättävien RF-kytkimien ohjaukset jääneet väärin päin, jonka johdosta ajurille täytyi muistaa antaa päätettävien porttien komennot erikseen.

Komentoja on kahdenlaisia, HIGH- ja LOW-tilojen kirjoitukseen. Ne ovat muotoa: `Close;a;b;c;d` ja `Open;a;b;c;d`, joissa a,b,c ja d edustavat Arduinon kytkettävien pinnien numeroita. Close-komennolla kirjoitetaan HIGH-tila ja Open-komennolla LOW-tila. Esimerkiksi komennolla `Close;32;34;36` kytketään Arduinon pinnit 32, 34 ja 36 HIGH-tilaan.

5.2.2 EEPROMin kirjoitus ja lukeminen

EEPROMin kirjoitusta ja lukemista varten oli olemassa jo valmiit VI:t, joten niiden yhdistäminen ajuriin oli riittävä toimenpide, jotta EEPROMin kirjoitus ja lukeminen toimivat. Kuvassa 27 on koodissa käytetty ali-VI, jossa EEPROMilta luetaan tietoja.

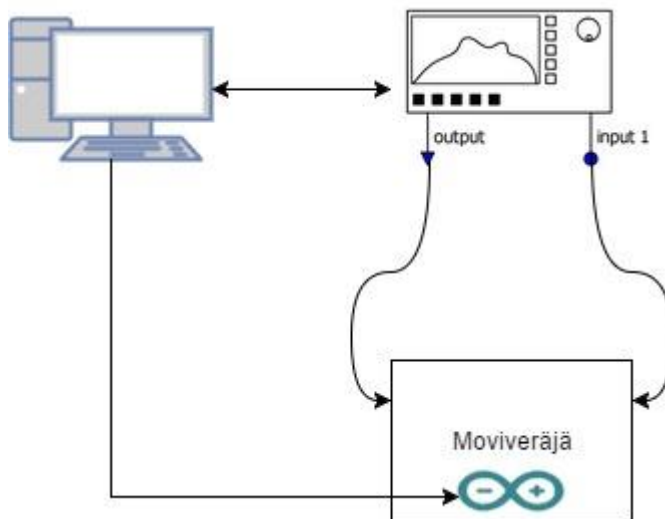


Kuva 27. EEPROMin lukemiseen käytetty ali-VI.

EEPROMin lukemisen hoitavan ali-VI:hin tuloina ovat I2C-väylän kanava, Arduinon osoite, LINX Resource -klusteri, luettavan muistipaikan aloitusosoitteen datatyyppi, luettavan muistipaikan aloitusosoite, luettava tavumäärä, error-klusteri sekä kommunikation aikakatkaisu. Ali-VI palauttaa LINX resource -klusterin, luetun datan ja error-klusterin.

5.3 Ensimmäiset mittaukset

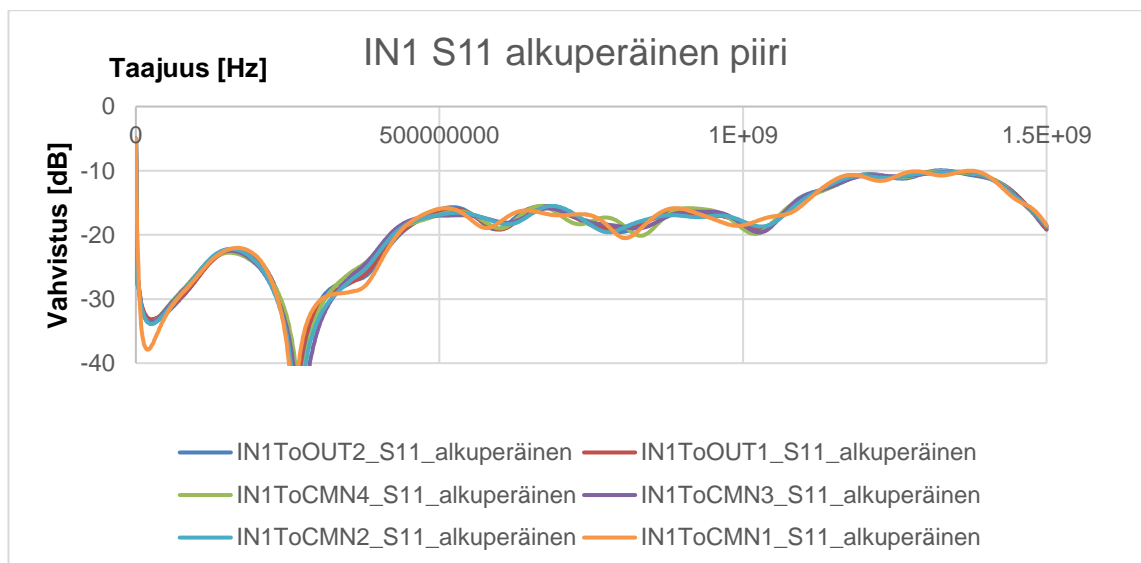
Mittauksissa käytettiin Keysightin E5061B piirianalysointia, jonka taajuusalue on 0,1—1500 MHz ja tehonkesto 20 dBm RF ja 30 V DC [22]. Piirianalysointia kalibroinnissa käytettiin Keysightin 85096C elektronista kalibrointimoduulia. Koska moniveräjässä on kolme 50 Ω :n porttia, oli mittauksissa käytettävä myös 75 \rightarrow 50 Ω sovittimia, käytössä oli E-Microwaven EMIMCT-5075-NMF sovittimia. Sovittimien vaikutus otettiin huomioon mittauksissa poistamalla niiden vaimennus mittaustuloksista. Piirianalysointia mittaustulosten määrä oli 1601, IF-kaistanleveys 4 kHz ja teho 0 dBm. Mittausalueena oli 0,3—1500 MHz. Kuvassa 28 on käytetty mittaasetelma.



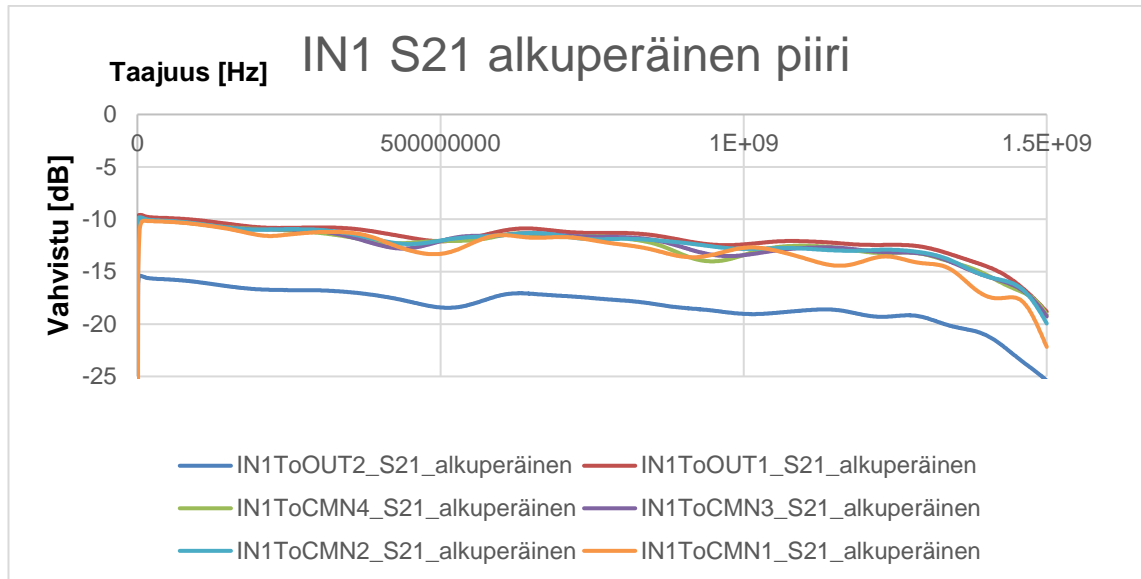
Kuva 28. Käytetty mittausasetelma.

Käytössä oli valmis LabVIEW:lla toteutettu mittausohjelma, joka poisti mittauksista sovitimien vaikutuksen ja tallensi tulokset automaattisesti.

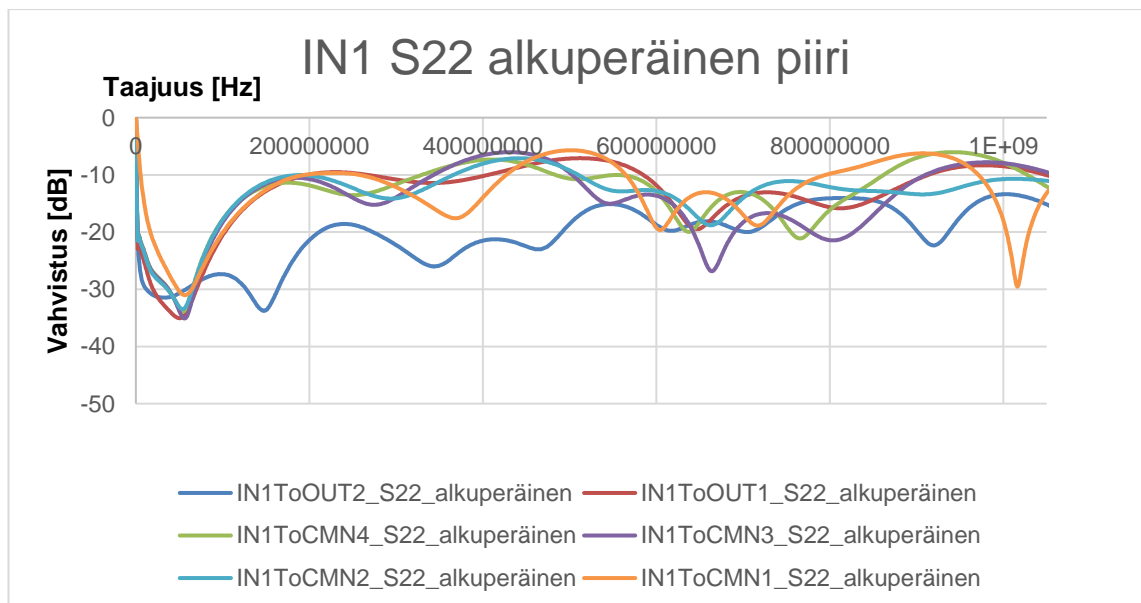
Alkuperäisen piirin portin IN1 mittaukset on esitetty kuvioissa 3–5.



Kuvio 3. Moniveräjän portin IN1 heijastusvaimennus S11 taajuuden funktiona.



Kuvio 4. Moniveräjän portin IN1 läpäisy taajuuden funktiona.



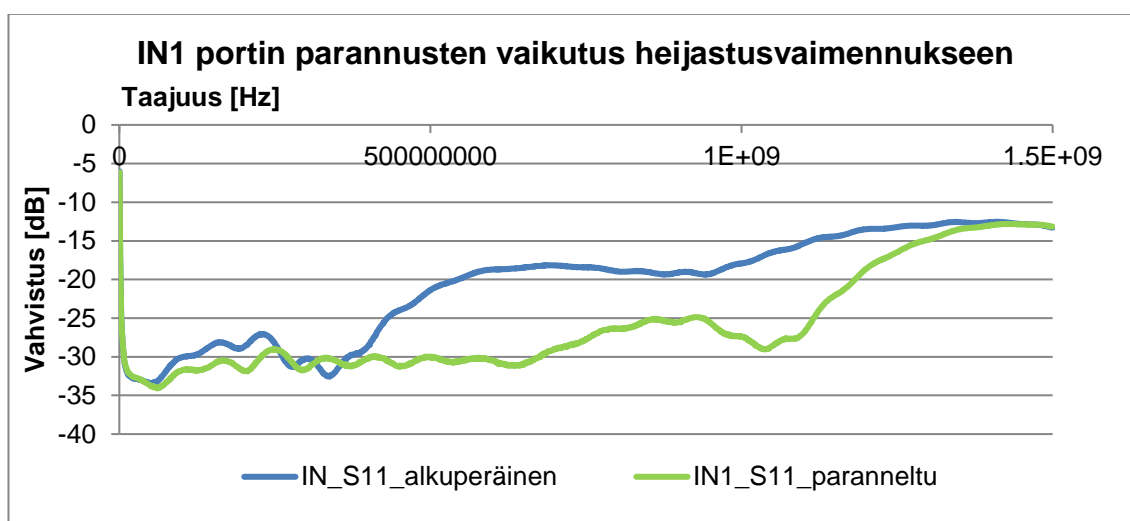
Kuvio 5. Moniveräjän portin IN1 heijastusvaimennus S22 taajuuden funktiona.

Tuloksista kävi ilmi, että moniveräjän läpäisy oli kohtalaisen hyvä, mutta heijastusvaimennukset eivät olleet vaatimusten tasolla. Läpäisyn aaltoilu johtui suurelta osin huonosta heijastusvaimennuksesta. Heijastusvaimennuksien parantamiseksi oli siis keksittävä muutoksia moniveräjään.

5.4 Moniveräjän parannuksia

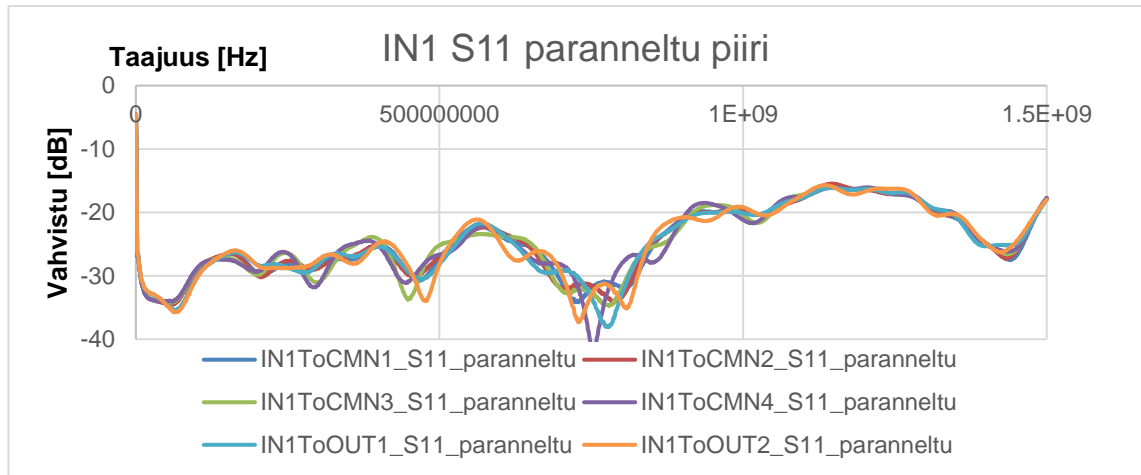
Suurimmat ongelmat tulivat heijastusvaimennuksista. Heijastusvaimennus johtuu epäjatkuuskohdasta siirtolinjassa, esimerkiksi epäsovituksista. Parannusmahdollisuuksia lähdettiin tutkimaan pienissä osissa. Aluksi tutkittiin ja paranneltiin porttien HV:ta päättämällä kyseisen portin linja 75 Ω :n vastuksella maihin ennen RF-kytkintä, joka yhdistäisi portin siirtolinjan sellaiseen siirtolinjaan, jota käyttää useampi portti. CMN1-portin tulossa oleva 48 V:n linja erotettiin signaalireitistä poistamalla niiden välillä oleva kuristin, jotta saatiin varmistettua kyseisen portin samankaltaisuus muiden CMN-porttien kanssa.

Moniveräjän paranneltuja tuloksia mitattiin Agilentin E5061A piirianalysointia, jonka taajuusalue on 0,1–1 500 MHz ja tehonkesto 20 dBm RF ja 30 V DC [23]. Kuviossa 6 on vertailtu yhden portin parannusten vaikutusta lähtötilanteeseen.

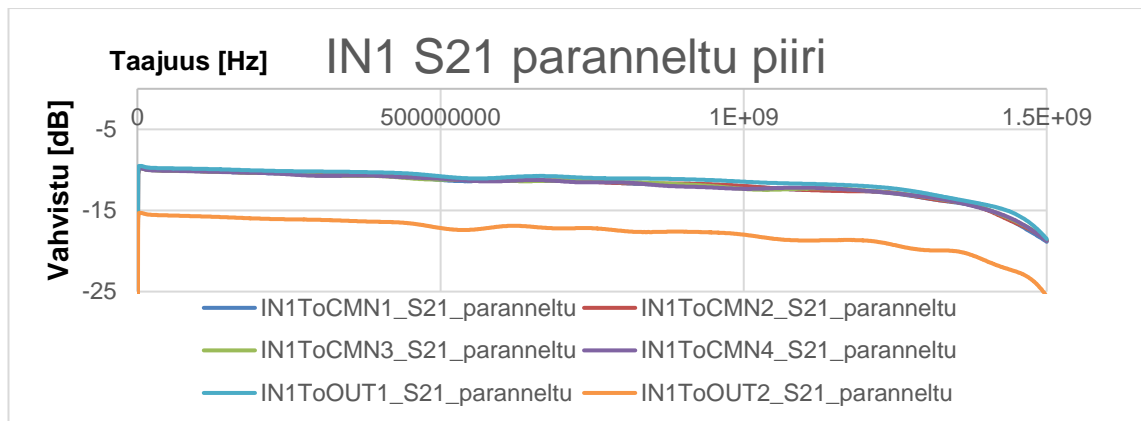


Kuvio 6. Moniveräjän portin IN1 parannusten vaikutus heijastusvaimennukseen.

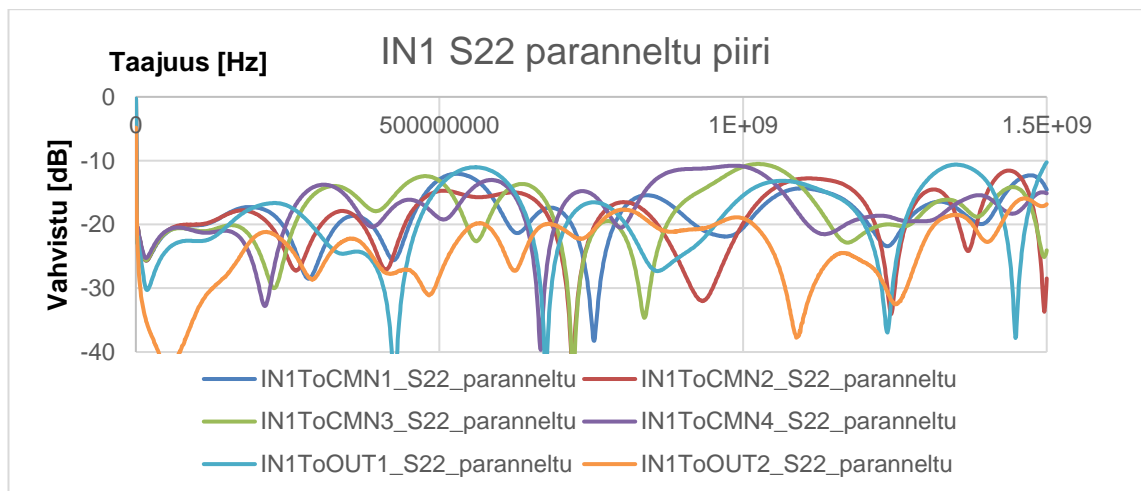
Parannuksien avulla porttien alkupään HV:t saatiin yli 20 dB:n, eli vain 1% lähteen tehosta heijastuu takaisin. Kyseessä oli kuitenkin vain kunkin portin signaalireitin alkupää, ei koko signaalireitti. Koko reitin parannuksia ei suoritettu, sillä piirilevyn layout tulee muuttumaan moniveräjän jatkokehityksen aikana. Kuvioissa 7–9 näkyy IN1-portin mittaustulokset portin parantelun jälkeen.



Kuvio 7. Parannellun piirin portin IN1 heijastusvaimennus S11.



Kuvio 8. Parannellun piirin portin IN1 läpäisy.

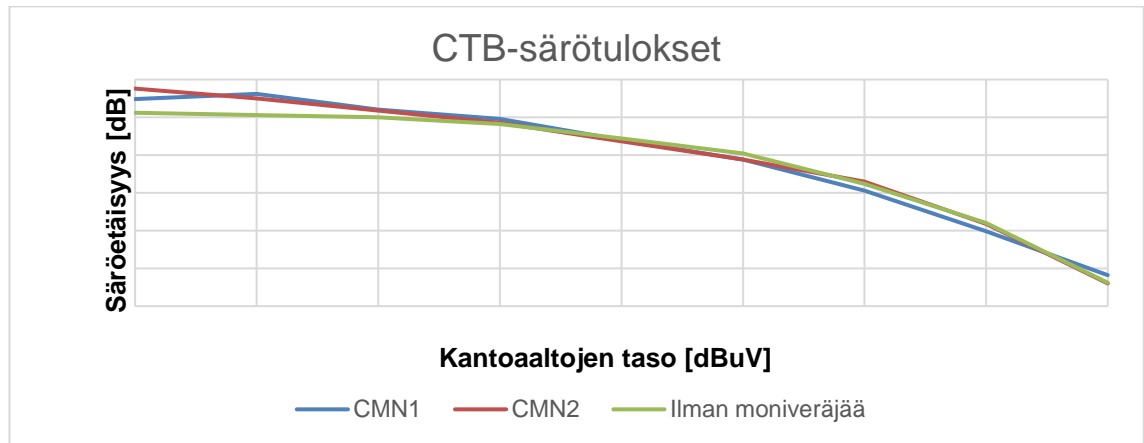


Kuvio 9. Parannellun piirin portin IN1 heijastusvaimennus S22.

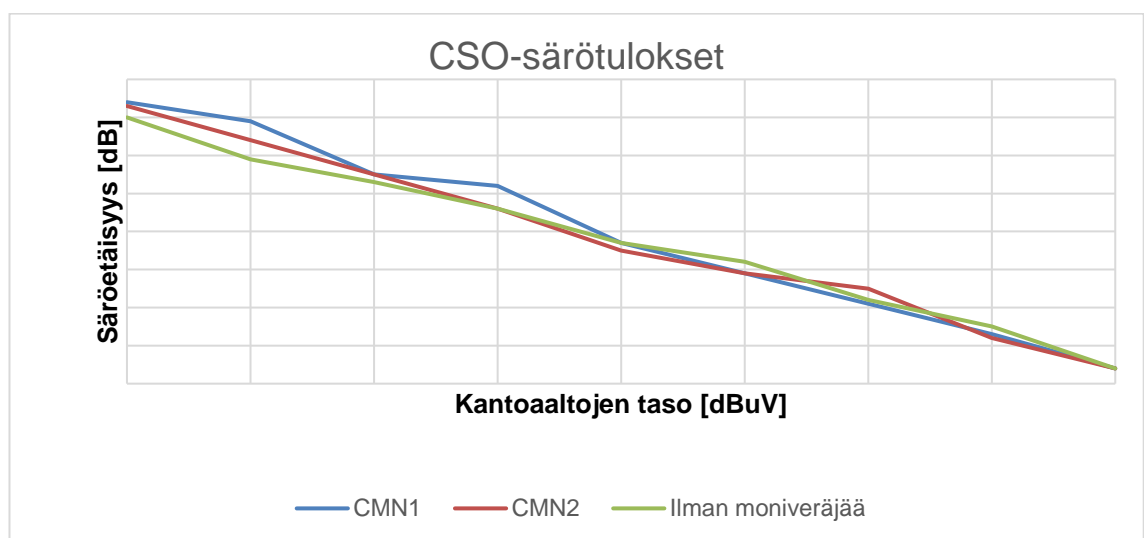
Tuloksista ilmeni, että porttien heijastusvaimennukset ja läpäisy paranivat jo vähäisillä muutoksilla. Läpäisy ei enää aaltoillut kuin vähän, johtuen paremmista heijastusvaimennuksista.

5.5 Moniveräjän särömittaukset

Moniveräjän särömittaukset suoritettiin eräällä Telesten vahvistimella, mittaamalla säröjen tasoa usealla eri taajuudella ja kantoaaltojen tasolla. Näin saatiin mahdollisimman kattavasti tuloksia moniveräjän säröominaisuuksista. Kuvioissa 10–11 on CTB ja CSO säröt eräillä taajuuksilla kahdella eri moniveräjän portilla.



Kuvio 10. Mitatut CTB-säröt.



Kuvio 11. Mitatut CSO-säröt.

Moniveräjän kanssa mitatut särötulokset olivat hyviä, tulokset eivät juuri muuttuneet ver-
ratessa tuloksiin ilman moniveräjää. CTB-mittausten alkupään parempi tulos mitatessa
moniveräjän avulla johtuu siitä, että särömittapaikan vahvistimelle asetetaan korkeampi
taso, johtuen moniveräjän vaimennuksesta.

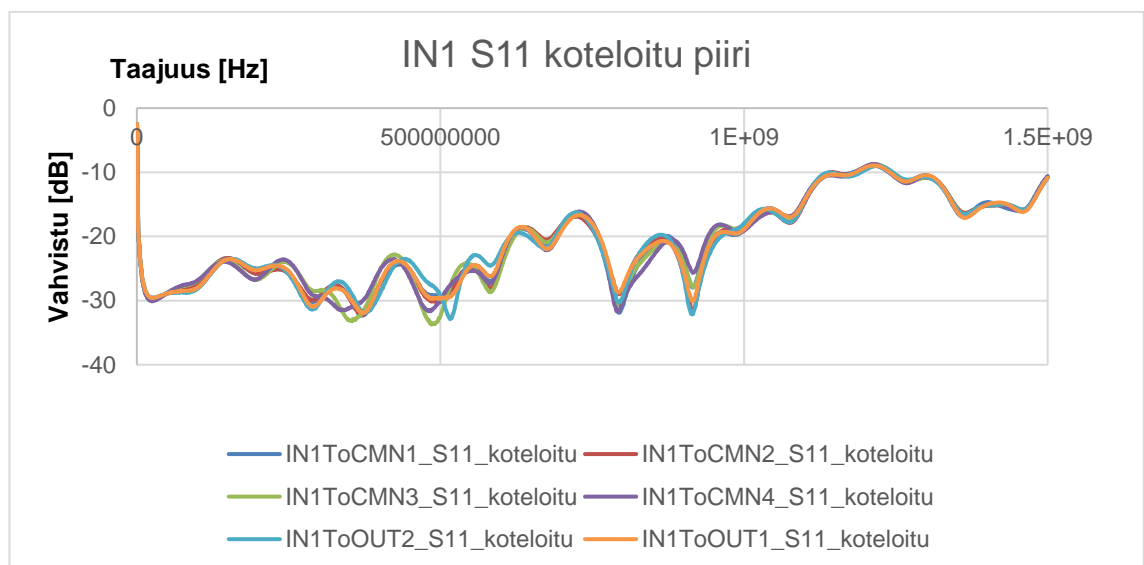
5.6 Moniveräjän kotelointi ja koteloinnin vaikutus mittaustuloksiin

Moniveräjän koteloksi valittiin jo aiemmin tuotannossa käytössä ollut kotelotyyppi, jonka
pystyy kätevästi sijoittamaan mittalaitetelineeseen. Kuvassa 29 on moniveräjän kotelon
etuosa, johon merkattu porttien nimet.

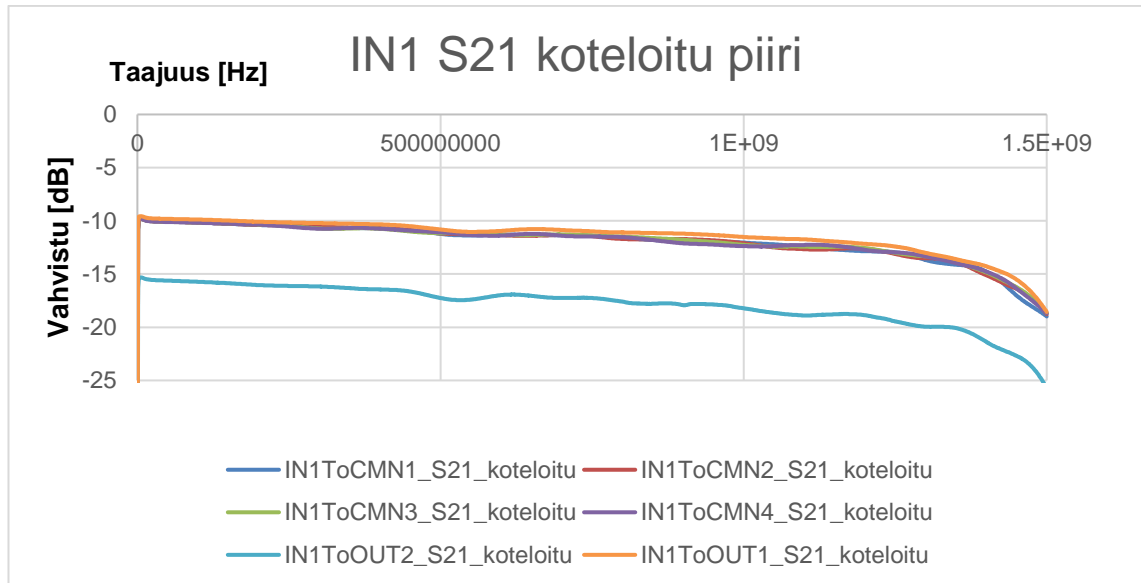


Kuva 29. Moniveräjän kotelon etuosa.

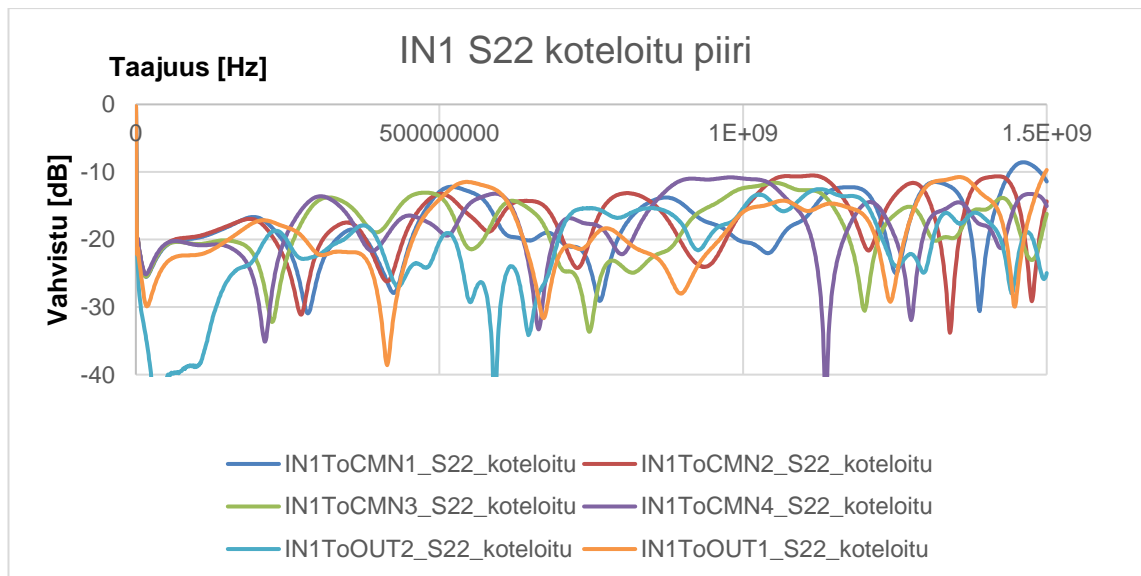
Koteloinnin valmistuttua moniveräjälle suoritettiin loppumittaukset, kuvioissa 12–14 on
IN1-portin tulokset. Kaikki loppumittaustulokset on liitteessä 1.



Kuvio 12. Koteloidun moniveräjän portin IN1 heijastusvaimennus S11.



Kuvio 13. Koteloidun moniveräjän portin IN1 läpäisy.



Kuvio 14. Koteloidun moniveräjän portin IN1 heijastusvaimennus S22.

Moniveräjän porttien liittimien koaksiaalikaapeleita joutui hieman lyhentämään, joka vaikuttaa mittaustuloksiin koteloinnin vaikutuksen lisäksi. Tulokset muuttuivat huonommiksi kuin ennen kotelointia, heijastusvaimennuksia on saatava paremmaksi moniveräjän seuraavaan versioon. Moniveräjä todettiin kuitenkin tarpeeksi valmiiksi, jotta sen voi ottaa tuotantoon testikäyttöön.

6 PÄÄTELMÄT

Työn suunnittelu ja toteutus onnistui hyvin, moniveräjän toiminta testattiin ja todettiin tarpeeksi hyväksi, jotta sen voi ottaa testikäyttöön tuotantoon. Haluttuja laatuvaatimuksia ei kuitenkaan täytetty kaikkia, varsinkin piirin heijastusvaimennukset tuottivat ongelmia. Piirilevyn suurta kokoa voi pienentää vaihtamalla Arduino Mega 2560:n Arduino Nanoon, jolloin on käytettävä I/O-levitintä pinnien riittämiseksi.

Vaikka joidenkin moniveräjän porttien HV:t saatiin vasta lähes halutulle tasolle, niitä täytyy pystyä parantamaan entisestään. Porttien heijastusvaimennuksia voi saada vielä paremmaksi vaihtamalla niihin juotetut, koaksiaalikaapelin päissä olevat liittimet, kiinteiksi. Summaimen 3DSS2 voi vaihtaa pintoliitosmalliseen summaimeen, joka ei tarvitse koaksiaalikaapeleita. Lisäksi piirilevyllä risteävät RF-linjat voi yrittää tehdä piirilevyn eri kerroksiin ja laittaa risteävään kohtaan kelan kompensoimaan kasvanutta kapasitanssia [24].

Särömittausten tulokset moniveräjän kanssa olivat hyviä ja se toimi odotusten mukaisesti sille asetetussa ympäristössä. Valitut RF-kytkimet osoittautuivat säröominaisuuksiltaan hyväksi, joten niitä suositellaan käytettäväksi myös moniveräjän jatkokehityksessä.

Ajuri toimi halutulla tavalla ja oli suunniteltu sille asetettujen vaatimusten mukaisesti. Sen sai suoraan yhdistettyä jo käytössä olleeseen, TestStandilla toteutettuun mittaussekvenssiin. Moniveräjä testauksen aikana kuitenkin tuli ilmi muutamia parannusmahdollisuuksia, jotka olisi hyvä toteuttaa ajurin seuraavaan versioon. Suurin muutos oli moniveräjän ohjausten topologian sijoittaminen.

Moniveräjän ohjauksien topologia toteutettiin ulkoiseen tekstitiedostoon, jonka ylläpitäminen saattaa olla hankalaa. Sen sijaan topologian voisi sijoittaa ajuriin, jolloin ajuri tulisi päivittää topologian muuttuessa, mutta tekstitiedostossa olevat käskyt pitäisivät aina paikkansa.

Moniveräjän kotelointi vaikutti toimivalta ratkaisulta, vaikka tulokset hieman huononivat sitä käytettäessä. Se oli helppo toteuttaa ja näytti siistiltä, lisäksi se mahtui kätevästi särömittapaikan mittalaitetelineeseen. Jos moniveräjän piirilevy kuitenkin muuttuu jatkokehityksen yhteydessä, voi olla tarpeen muuttaa myös kotelointia.

7 YHTEENVETO

Työssä suunniteltiin ja toteutettiin särömittapaikan siirtolinjojen automaattinen ohjausjärjestelmä Teleste Oyj:lle. Särömittapaikan mittaukset olivat automatisoitu jo aiemmin, mutta työn tekijä joutui kuitenkin siirtelemään siirtolinjoja kesken mittauksien. Tämä toiminta haluttiin automatisoida työn nopeuttamiseksi ja virhekytkentöjen välttämiseksi.

Aiemmin käytössä oli ollut Agilentin valmistama multiportti, joka ei kuitenkaan toiminut vaatimusten mukaisesti kun mitattavien laitteiden taajuusalueen ylärajaa kasvatettiin aiemmasta 1 GHz:stä 1,2 GHz:iin. Muita kaupallisia valmiita ratkaisuja ei ollut, joten oli luontevaa suunnitella ja räätälöidä uusi laite omien tarpeiden mukaan. Tämä myös mahdollistaa laitteen jatkokehityksen, esimerkiksi porttien lisäämiseksi tai toiminnallisen taajuusalueen kasvattamiseksi. Laitteen ohjaukseen valittiin Arduino Mega 2560, koska se on halpa, helppokäyttöinen ja varmatoiminen. Lisäksi käyttöön valittiin LabVIEW-ohjelmointiympäristö, joka oli jo entuudestaan käytössä ja siitä on Telestellä paljon asiantuntemusta.

Ensimmäinen versio moniveräjästä ei saavuttanut sille asetettuja vaatimuksia kokonaan. Sen vaimennus oli tarpeeksi pieni, mutta porttien heijastusvaimennukset olivat liian suuria. Siihen täytyi tehdä muutoksia, jotta se olisi mahdollista ottaa käyttöön tuotantoon. Heijastusvaimennukset tuottivat suurimmat ongelmat, joten niiden parannukset olivat välttämättömiä. Ajuri toimi halutulla tavalla, eikä tarvinnut suuria muutoksia.

Parannellun piirilevyn heijastusvaimennukset ja vaimennus olivat kohtuullisen hyviä. Särötulokset olivat hyviä, moniveräjän kanssa tehdyt särömittaukset eivät tuottaneet huonompia tuloksia kuin ilman moniveräjää. Tämän työn valmistuttua moniveräjä otetaan testikäyttöön tuotantoon ja sitä jatkokehitetään testikäytössä ilmenevien tarpeiden mukaan, sekä päätelmissä esitettyjen parannusehdotusten pohjalta.

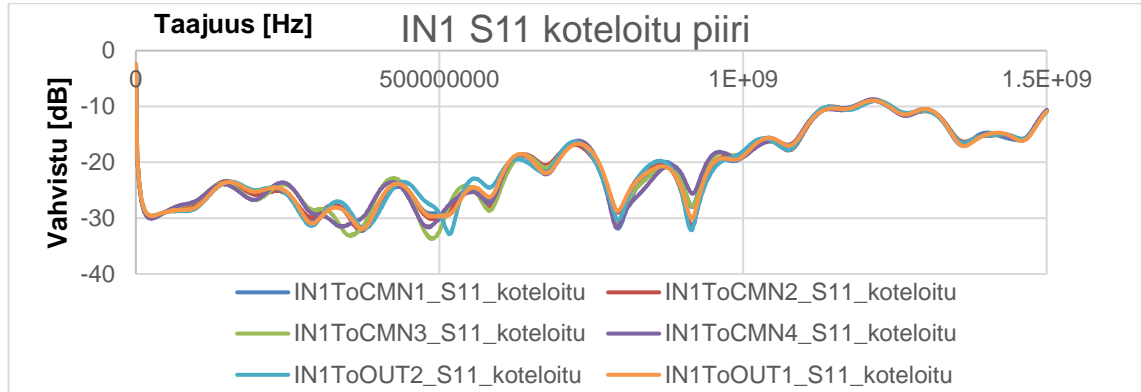
LÄHTEET

- [1] Bowick, Chris, Blyler, John ja Ajluni, Cheryl, *RF Circuit Design*. 2nd ed. Oxford: Elsevier Science, 2008. 243 s. ISBN: 978-0-7506-8518-4
- [2] Pulkkinen, P., *EMC-laboratorion kytkentäyksikkö*, opinnäytetyö, Kajaanin ammattikorkeakoulu, 2012, 25 s.
- [3] Liker, Jeffrey K., *Toyotan tapaan*. Suomentanut Niemi, Marko. Helsinki: readme.fi, 2010. 323 s. ISBN: 976-952-220-226-0
- [4] National Instruments, "What is LabVIEW?", [www-dokumentti]. Saatavilla: <http://www.ni.com/newsletter/51141/en/> (luettu: 11.3.2017)
- [5] Essick, John, *Hands on introduction to LabVIEW for scientists and engineers*. 3rd ed. New York: Oxford University Press, 2016. 668 s. ISBN: 978-0-19-021189-9
- [6] National Instruments, "LINX by Digilent/Labview Makerhub", [www-dokumentti]. Saatavilla: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/212478> (luettu: 11.3.2017)
- [7] GitHub, "LabVIEW hacker LINX", [www-dokumentti]. Saatavilla: <https://github.com/MakerHub/LINX> (luettu: 12.3.2017)
- [8] National Instruments, "What is TestStand?", [www-dokumentti]. Saatavilla: <http://www.ni.com/teststand/whatis/> (luettu: 12.3.2017)
- [9] Arduino, "What is Arduino?", [www-dokumentti]. Saatavilla: <https://www.arduino.cc/en/Guide/Introduction> (luettu 21.2.2017)
- [10] Arduino, "Frequently asked questions", [www-dokumentti]. Saatavilla: <https://www.arduino.cc/en/Main/FAQ> (luettu 21.2.2017)
- [11] Ulaby, Fawwaz T., Michielssen, Eric ja Ravaoli, Umberto, *Fundamentals of Applied Electromagnetics*. 6th ed. Noida: Pearson India Education Services Pvt. Ltd, 2015. 486 s. ISBN: 978-93-325-3514-5
- [12] Rohde & Schwarz, "CSO, CTB & XMOD characterisation of CATV line extenders with the aid of vector signal generators", [www-dokumentti]. Saatavilla: https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/00aps_undefined/RAC-0605-016.pdf (luettu: 28.4.2017)
- [13] Carlson, A. Bruce ja Crilly, Paul B., *Communication systems*. 5th ed. New York: McGraw Hill Education (India) Private Limited, 2014. 924 s. ISBN: 978-0-07-132117-4
- [14] Arduino, "Arduino MEGA 2560 & Genuino MEGA 2560", [www-dokumentti]. Saatavilla: <https://www.arduino.cc/en/Main/arduinoBoardMega2560> (luettu: 21.2.2017)
- [15] "PE42722", Peregrine semiconductor, San Diego, Kalifornia, USA. 2014.
- [16] Peregrine semiconductor, "UltraCMOS technology platform, benefits of UltraCMOS technology", [www-dokumentti]. Saatavilla: <http://www.psemi.com/technologies/ultracmos-technology-platform/benefits-of-ultracmos-technology> (luettu: 20.2.2017)
- [17] Peregrine semiconductor, "UltraCMOS technology platform", [www-dokumentti]. Saatavilla: <http://www.psemi.com/technologies/ultracmos-technology-platform> (luettu 20.2.2017)

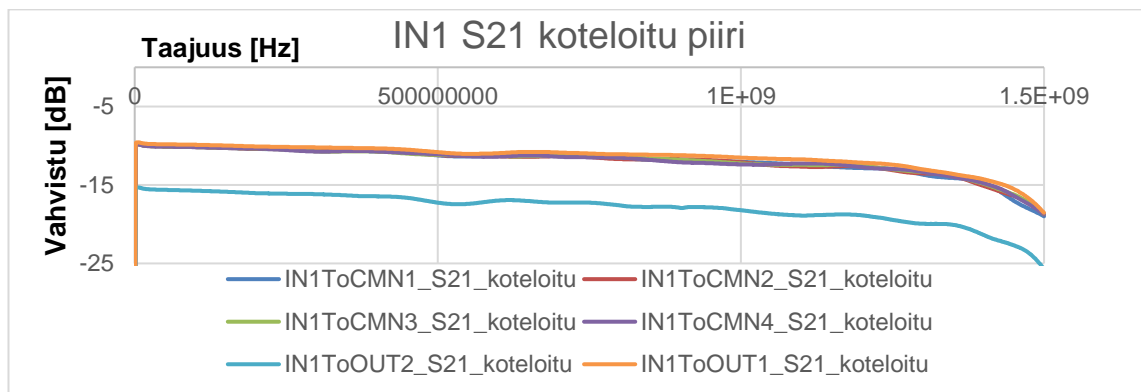
- [18] Kirkbribe, D., sähköpostikeskustelu, 23.2.2017.
- [FT232R] "FT232R USB UART IC", Future Technology Devices International Ltd., Glasgow, UK. 2015.
- [19] "3D-line", Teleste Corporation, Littoinen, Suomi. 2014.
- [20] *IEC 60063, Preferred number series for resistors and capacitors*. International electrotechnical commission IEC, Geneve, 1963.
- [21] Stolyarenko, Alexey A.; Rubanovich, Michael G.; Razinkin, Vladimir P.; Khrustalyov, Vladimir A. ja Aubakirov, Konstantin Y., "Simulation of 50 to 75 Ω impedance transformer with losses in the frequency band up to 3.5 GHz for a 50 W input power", *16th International conference of young specialists on micro/nanotechnologies and electron devices*, 2015
- [22] "Keysight E5061B", Keysight Technologies, Santa Rosa, Kalifornia, Yhdysvallat. 2016.
- [23] "Agilent ENA-L RF Network Analyzers", Agilent Technologies, Santa Clara, Kalifornia, Yhdysvallat. 2012.
- [24] Rintalaulaja, M., sähköpostikeskustelu, 15.5.2017.

Liitteet

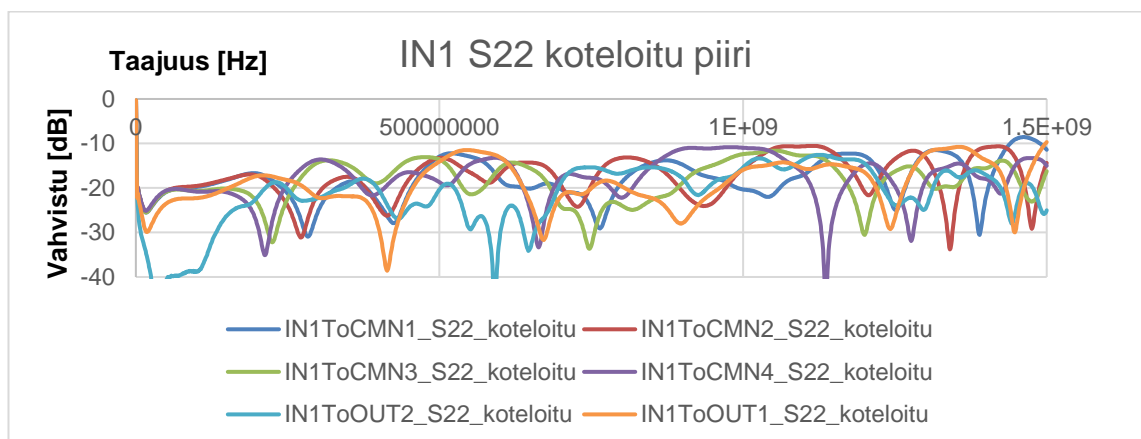
Moniveräjän loppumittausten tulokset



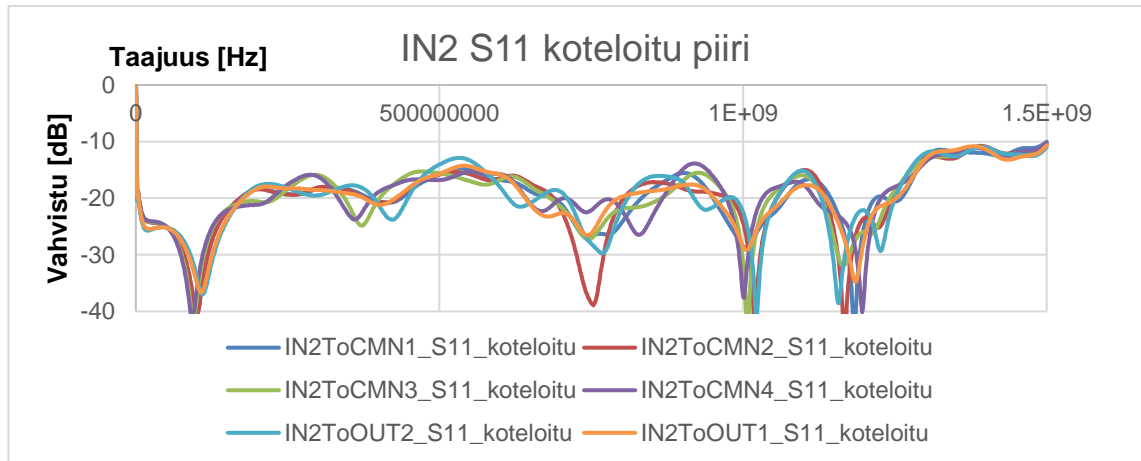
Kuvio 15. Moniveräjän portin IN1 S11, loppumittaus.



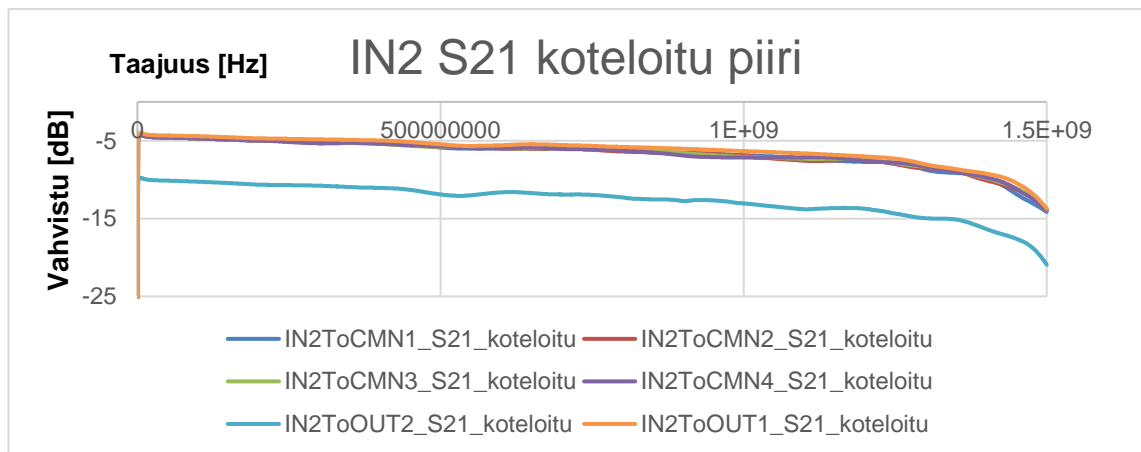
Kuvio 16. Moniveräjän portin IN1 S21, loppumittaus.



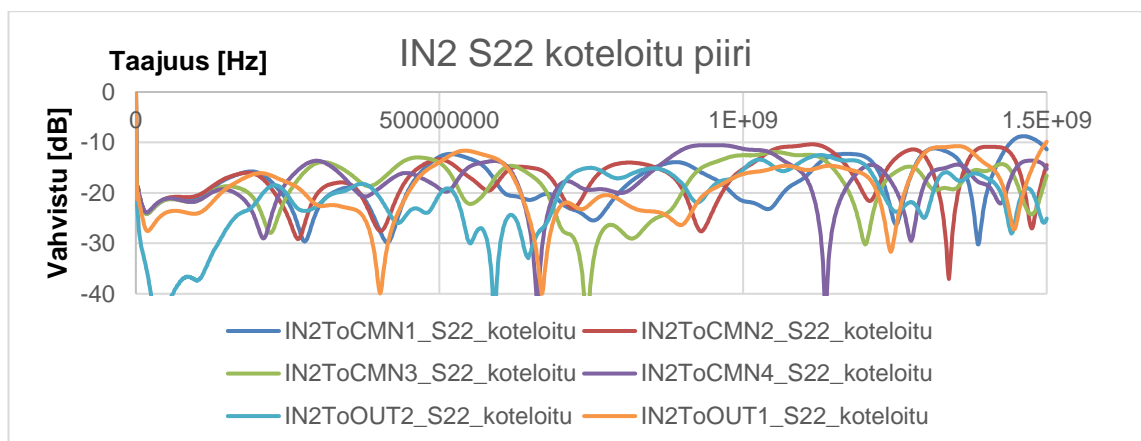
Kuvio 17. Moniveräjän portin IN1 S22, loppumittaus



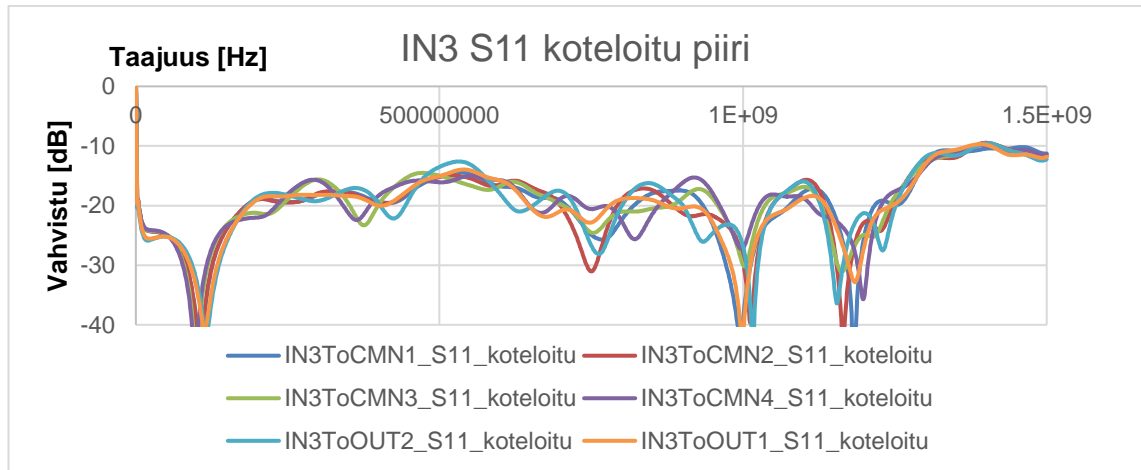
Kuvio 18. Moniveräjän portin IN2 S11, loppumittaus.



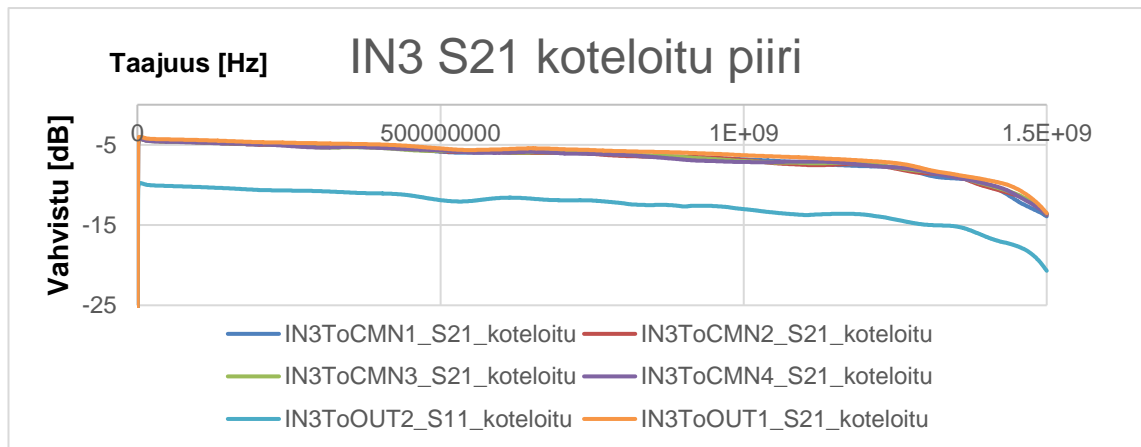
Kuvio 19. Moniveräjän portin IN2 S21, loppumittaus.



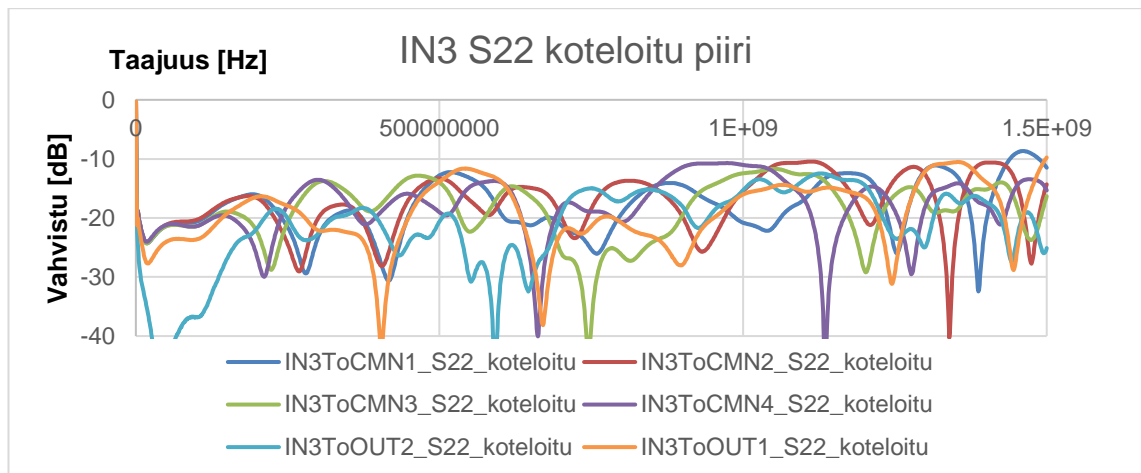
Kuvio 20. Moniveräjän portin IN2 S22, loppumittaus.



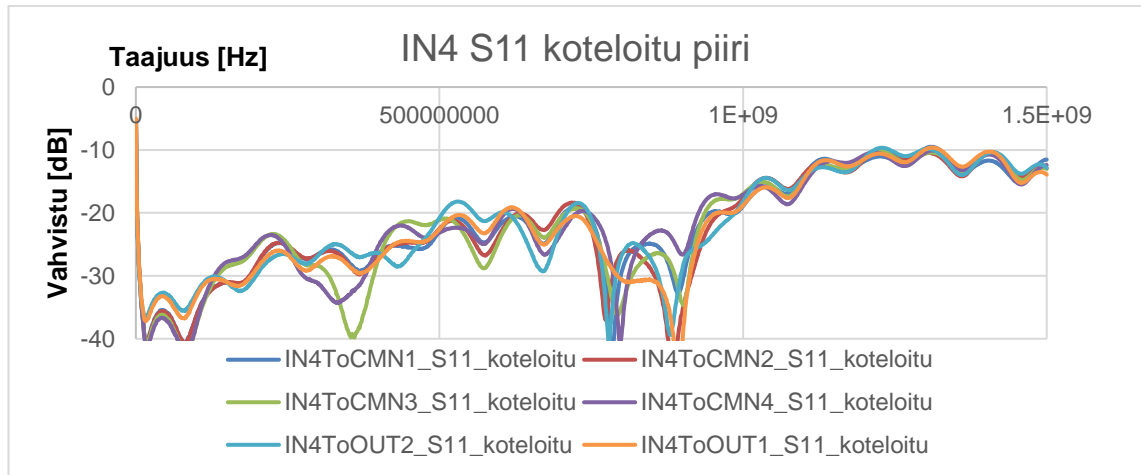
Kuvio 21. Moniveräjän portin IN3 S11, loppumittaus.



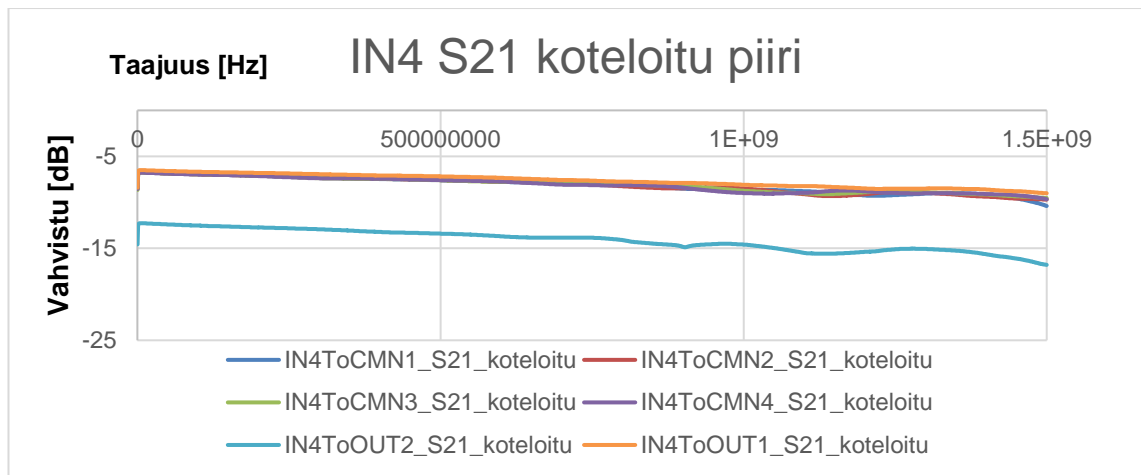
Kuvio 22. Moniveräjän portin IN3 S21, loppumittaus.



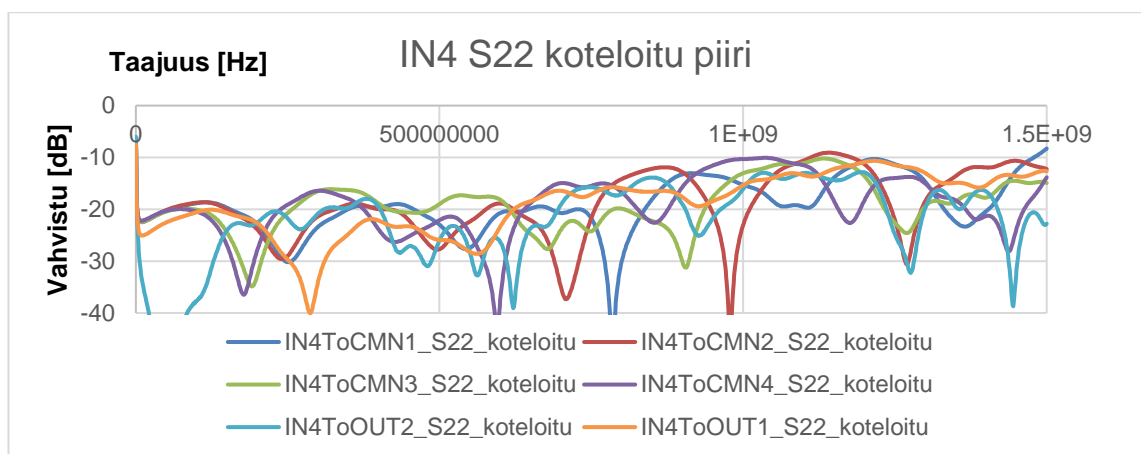
Kuvio 23. Moniveräjän portin IN3 S22, loppumittaus.



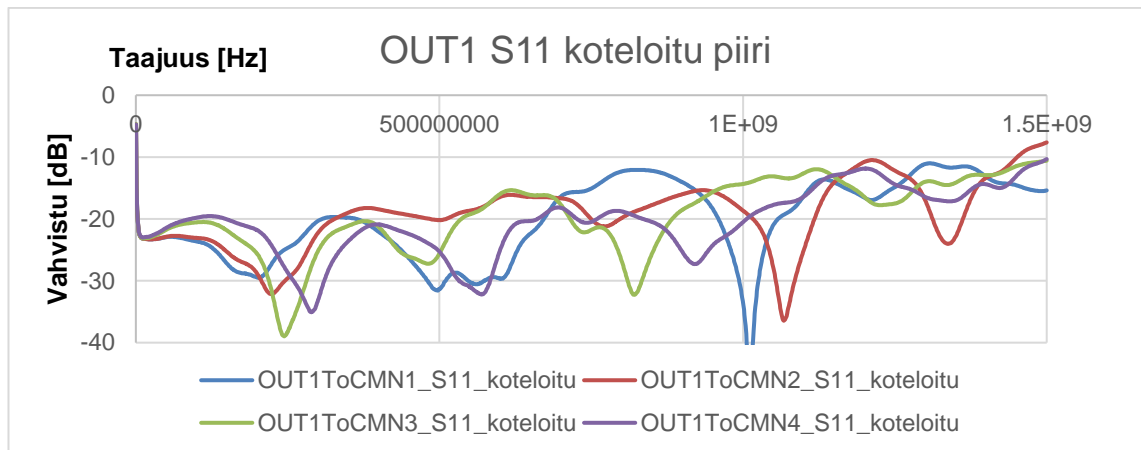
Kuvio 24. Moniveräjän portin IN4 S11, loppumittaus.



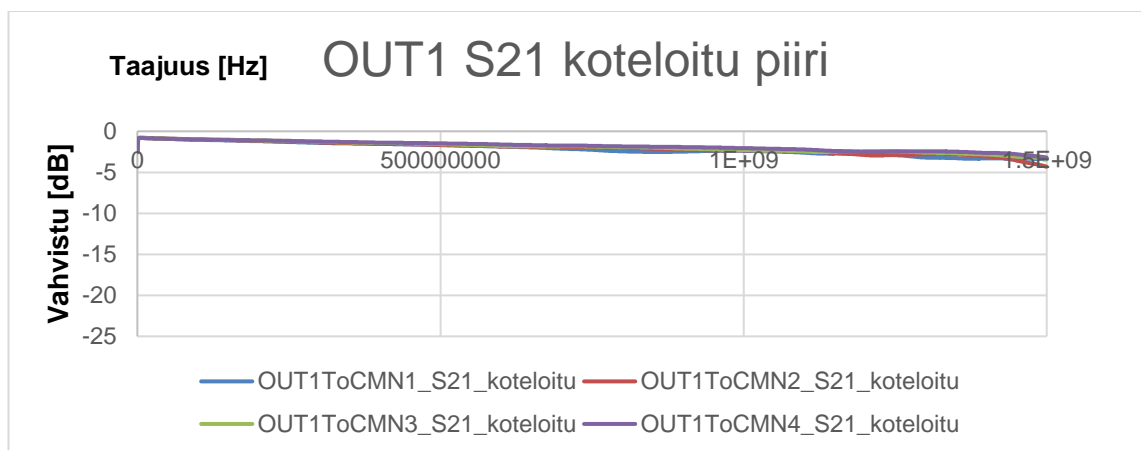
Kuvio 25. Moniveräjän portin IN4 S21, loppumittaus.



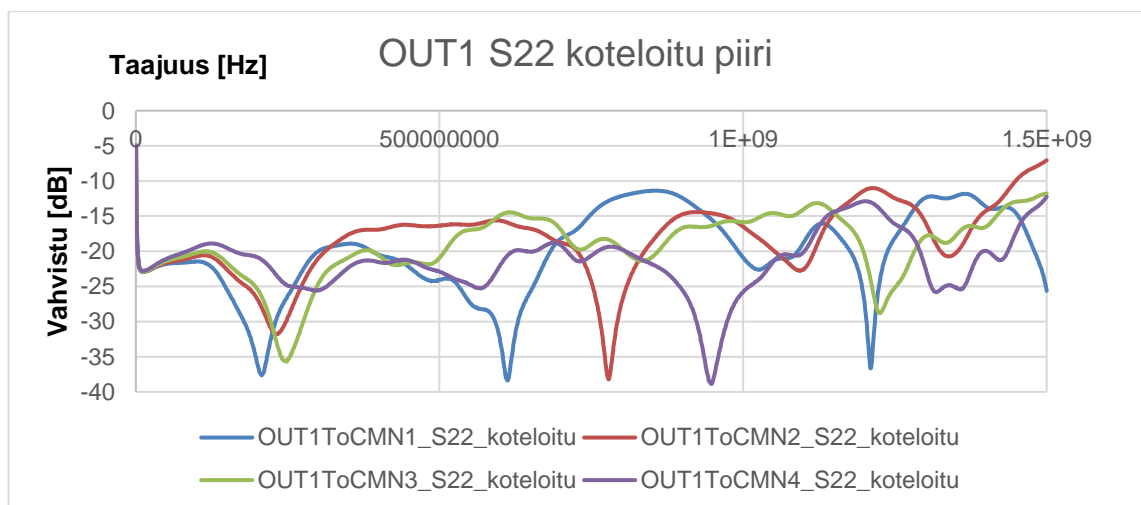
Kuvio 26. Moniveräjän portin IN4 S22, loppumittaus.



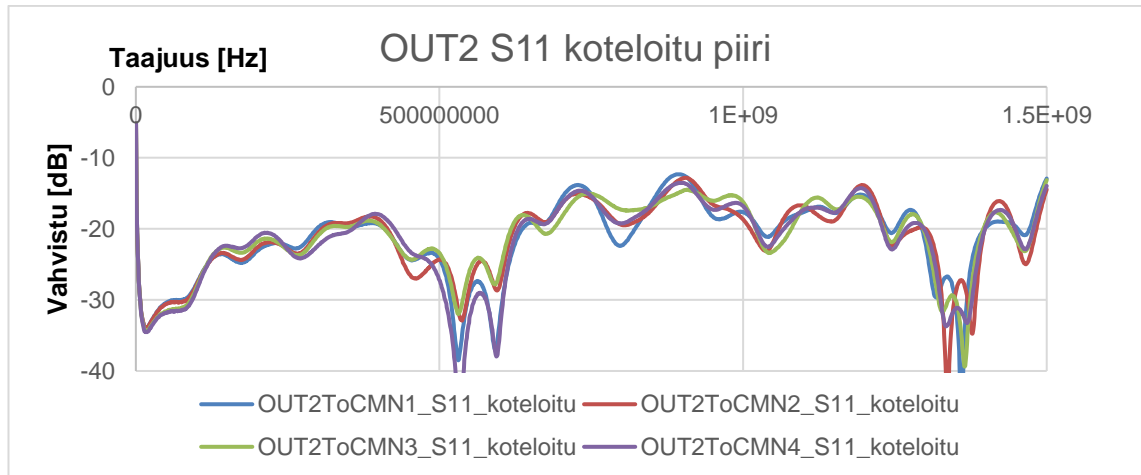
Kuvio 27. Moniveräjän portin OUT1 S11, loppumittaus.



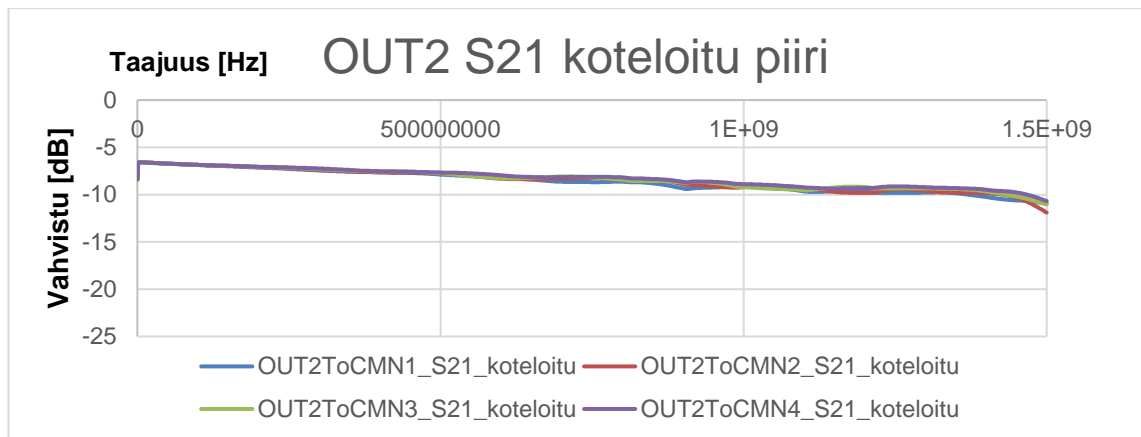
Kuvio 28. Moniveräjän portin OUT1 S21, loppumittaus.



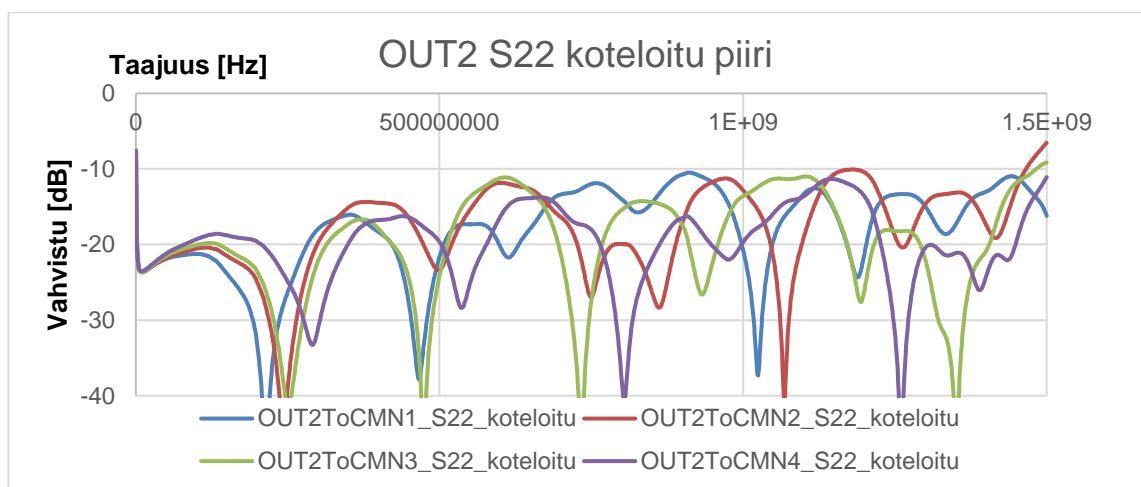
Kuvio 29. Moniveräjän portin OUT1 S21, loppumittaus.



Kuvio 30. Moniveräjän portin IN1 S11, loppumittaus.



Kuvio 31. Moniveräjän portin IN1 S11, loppumittaus.



Kuvio 32. Moniveräjän portin IN1 S11, loppumittaus.